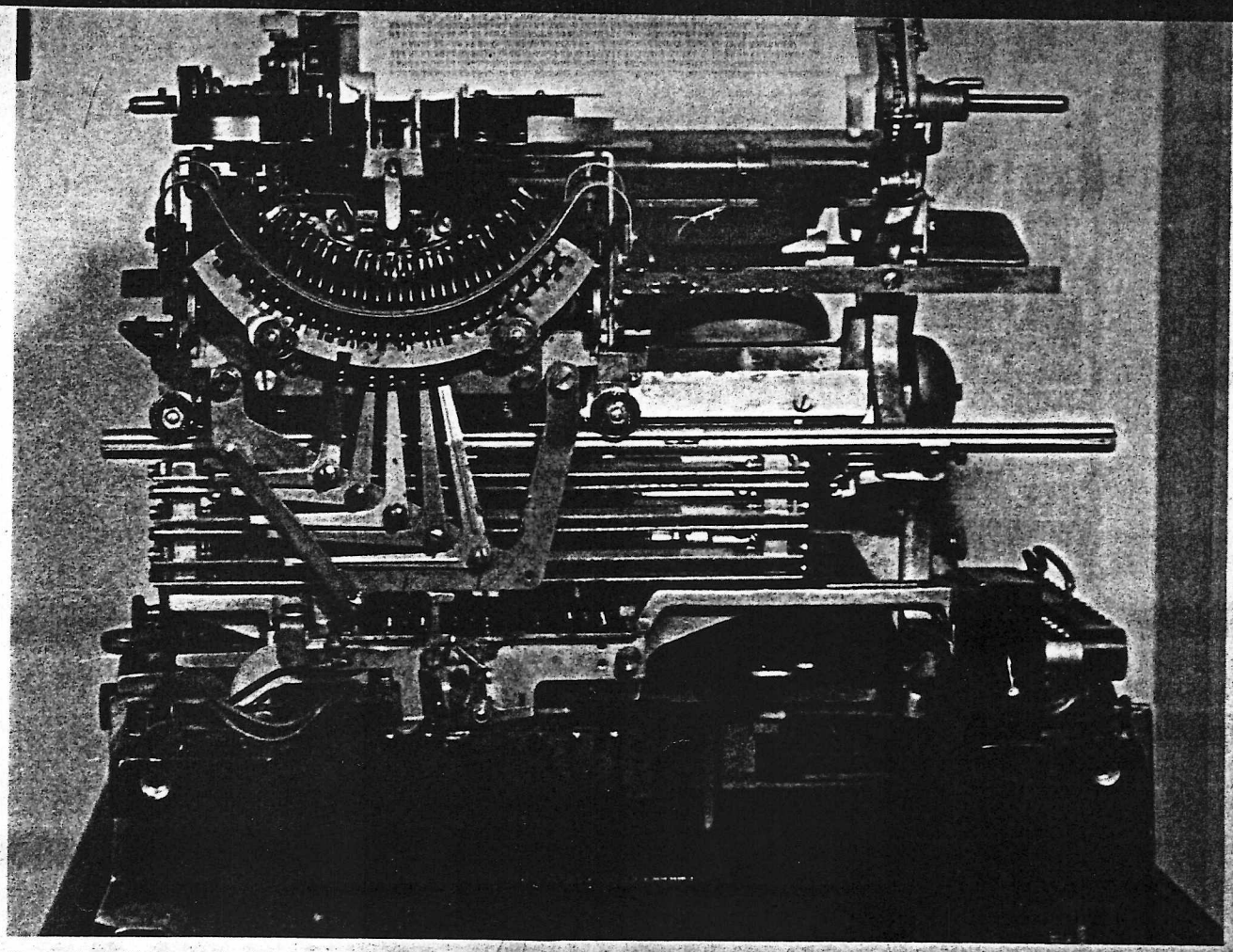
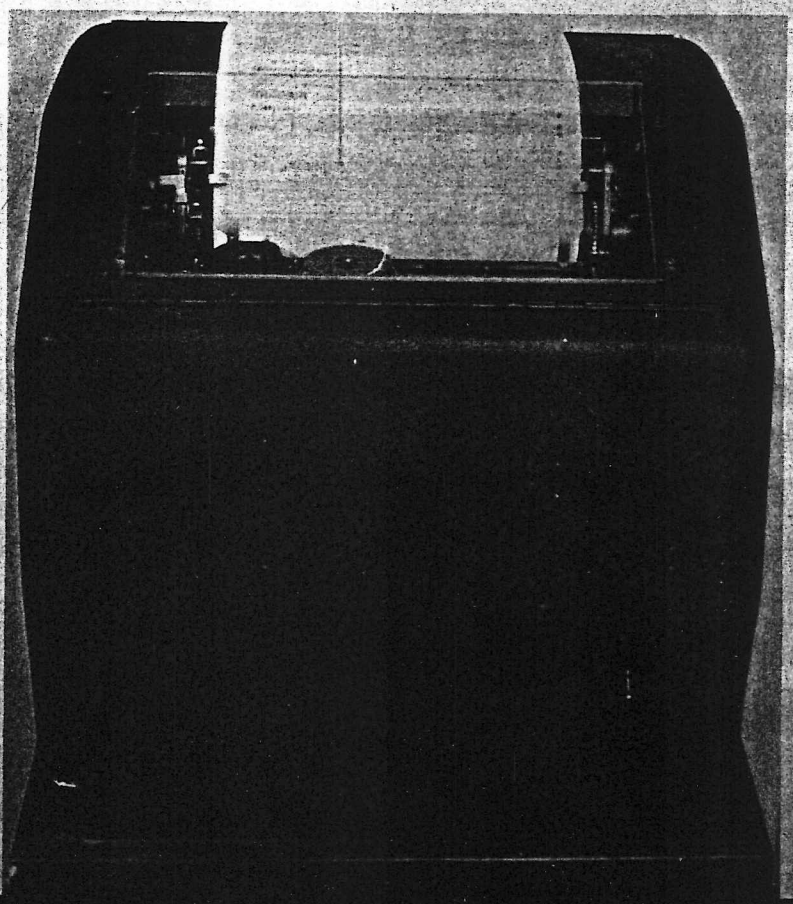


Photo 1: A Teletype Model 15 RO printer. This "receive only" model is shown with cover removed (above) and with the cover in place.



A Guide

Michael S McNatt
4658 E 57 St
Tulsa OK 74135

Are you wondering why you should consider yet another peripheral for your system when you may already have a nice, quiet video display and even a panel of Das Blinkenlights? Consider the following excellent quote by Robert H Cushman, one of the editors of *EDN* magazine, who has written several articles on designing with microprocessors:

Never Underestimate the Value of Hard Copy

Working at the Teletype is a superior way to learn . . . software and, later, to put the microprocessor to use . . .

	Bit Numbers 5 4 3 2 1	Lower (Letters) Case	Upper (Figures) Case Communications (Standard)	Bell System (TWX) Stock Market	Weather Code
0	0 0 0 0 0	Blank	Blank	Blank	—
1	0 0 0 0 1	E	3	3	3
2	0 0 0 1 0	Line Feed	Line Feed	Line Feed	Line Feed
3	0 0 0 1 1	A	–	–	↑
4	0 0 1 0 0	Space	Space	Space	Space
5	0 0 1 0 1	S	' (Apos)	' (Apos)	Bell
6	0 0 1 1 0	I	8	8	8
7	0 0 1 1 1	U	7	7	7
8	0 1 0 0 0	Car Ret	Car Ret	Car Ret	Car Ret
9	0 1 0 0 1	D	$ WRU	$	↗
10	0 1 0 1 0	R	4	4	4
11	0 1 0 1 1	J	Bell	Bell	↙
12	0 1 1 0 0	N	,	7/8	○
13	0 1 1 0 1	F	!	1/4	→
14	0 1 1 1 0	C	:	1/8	●
15	0 1 1 1 1	K	(	1/2	←
16	1 0 0 0 0	T	5	5	5
17	1 0 0 0 1	Z	+	"	+
18	1 0 0 1 0	L	)	3/4	↖
19	1 0 0 1 1	W	2	2	2
20	1 0 1 0 0	H	#	#	↓
21	1 0 1 0 1	Y	6	6	6
22	1 0 1 1 0	P	0	0	0
23	1 0 1 1 1	Q	1	1	1
24	1 1 0 0 0	O	9	9	9
25	1 1 0 0 1	B	?	5/8	○
26	1 1 0 1 0	G	&	&	↘
27	1 1 0 1 1	Figures	Figures	Figures	Figures
28	1 1 1 0 0	M	.	.	.
29	1 1 1 0 1	X	/	/	/
30	1 1 1 1 0	V	=	3/8	○
31	1 1 1 1 1	Letters	Letters	Letters	Letters

Table 1: A listing of the various Baudot key code assignments. The weather code column, taken from the Reference Data for Radio Engineers, *fifth edition, pages 35 to 38, table 22, includes wind direction and cloud cover symbols. In the communications column, the letters "WRU" stand for "Who are you?"*

to Baudot Machines:

Part 1, Description of Available Devices

The hardcopy record provided by the Teletype printer proved unexpectedly useful. Sample programs and their execution results . . . could be reviewed many days later, often clearing up today's cause of confusion . . .

Hardcopy records are especially helpful during this learning process because when one attacks some new feature of programming a particular machine, he or she invariably begins by probing and experimenting, making many false tries. At that instant, all inconclusive attempts seem like a complete waste of time. But later, as you advance, copies of your first attempts enable you to go back and explain earlier mysteries. Psychologically this is most comforting, because it wipes away the frustations that invariably build up when neither you nor the computer appear to understand each other.

EDN tears off the 6 foot or so strip of Teletype paper that results from the typical bout with the computer, folds it accordionstyle, and files it. To make our reference complete we start each session by entering the day's date and the session's objective, and end it with a summary of what was, or wasn't, accomplished.

The compact hardcopy printout that you get is a much more useful debugging tool than the transient

Photos 1-5 courtesy Donald Clum. Photo 6 courtesy Robert W Cook.

displays of console lamps, or the temporary frames of similar video readouts. You can study this hardcopy at leisure, make notes on it, and then file it away for future reference.

All these practical virtues of performing program development through typewriter-like terminals have been known to computer types for two decades. These old timers are amazed to learn that it is taking the newcomers to the computer world so long to catch up.

["How Development Systems Can Speed Up the μP Design Process," *EDN*, April 20 1976, pages 63-72.]

What Kinds of Baudot Machines are Available?

Perhaps the best way to answer the question of what's around is to catalogue each machine briefly, starting with the Teletype Corp models and following with units by other manufacturers such as Kleinschmidt, Creed, etc. In these descriptions, I've included photographs of many, but hardly all, of the typical machines. Most of these units will be found with nameplates and number plates identifying the model and manufacturer (if you purchase a whole machine).

Model 11

The first machine introduced with the name Teletype, in 1921. Used a type wheel and printed on paper tape at 40 words per minute.

Teletype Model 12

Introduced in 1922, about 12,000 of these units were sold. Practically all Model 12s had been removed from service by the 1940's, which makes it a real antique. The unit consisted of a Baudot page printer and a keyboard, and sounded like a cement mixer. The electrical noise or hash generated by these machines would require extensive shielding and cables for use with a microprocessor. Give it to a museum if you find one...

Teletype Model 14

This series of Baudot paper tape readers and punches were manufactured beginning in 1925, with about 57,000 Model 14s of all types being sold. The Model 14 series consists of:

- Model 14TD, or "transmitter distributor" reads the paper tape and transmits serial data on a current loop. Speeds of 60, 66, 75 or 100 words per minute can be selected by installing the appropriate set of gears. Easily modified for parallel output. Price range from $20 to $150.
- Model 14 RO "Reperforator" or receiver, punches paper tape from serial data received from a current loop. In 1940 a typing reperforator was introduced which also prints on the tape, but the printing is several characters behind the punches. Speeds are 60, 66, and 75 words per minute. Price range from $10 to $50.

Beware: Model 15s May be Hazardous to Your Health!

If you are considering a Teletype Model 15 or similar surplus page printer, be forewarned: Not only is the Model 15 obsolete (Teletype Corp no longer manufactures parts), but:

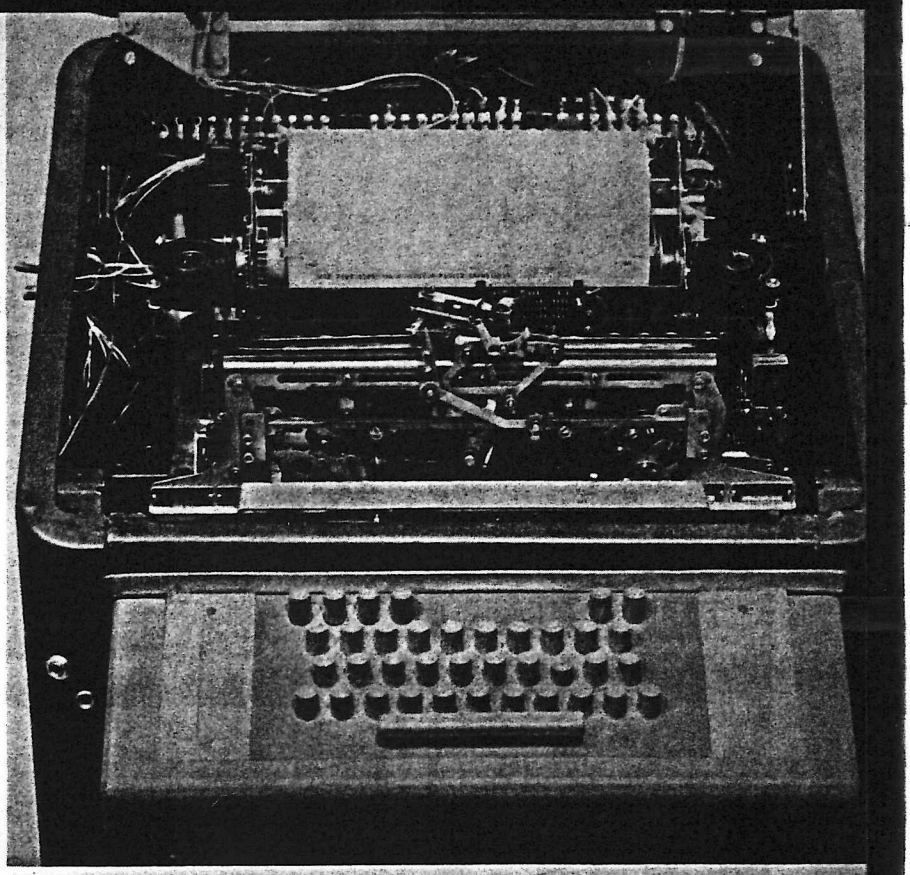
- *It's a rude, ugly mechanical monstrosity next to LED readouts, shiny microprocessor cabinetry, and solid-state keyboards.*
- *It's a veritable assault of clanking, grinding, and roaring among the silent flash of panel lights, video displays, and the barely audible hum of power supply transformers.*
- *Foul smelling oil and burning grease pour out of its bowels to drown the clean, woodsy smell of new programming books and operating manuals.*
- *It crawls along at a yawning, hair pulling, nail biting 60 to 100 words per minute, while a blazing processor hurries along wasting thousands of cycles between character outputs.*

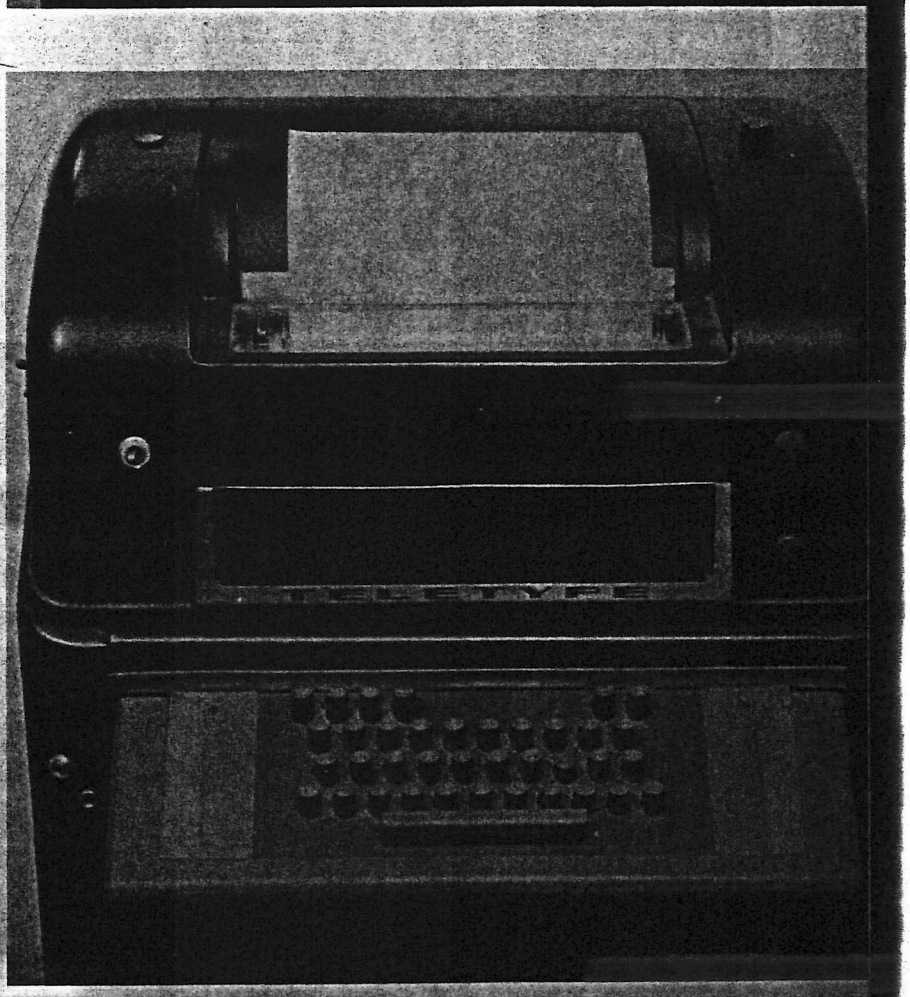
But a Model 15 hardcopy page printer will cost between $35 and $100 depending on your resourcefulness, and luck.

Table 2: Physical characteristics of the four common teleprinter speeds for Baudot machines. In many cases, the same machine can be run at several different speeds simply by changing gears.

Words Per Minute	Data Rate	Characters Per Second	Bit Pulse Width	Stop Pulse Width	Operations Per Minute
60	45 bps	6	22 ms	31 ms	368
66	50 bps	6.6	20 ms	30 ms	400
75	57 bps	7.5	18 ms	25 ms	460
100	75 bps	10	13.5 ms	19 ms	600

Photo 2: The author's Model 28 KSR Teletype machine. Note the much more compact mechanism (above) of this model compared to a similar view of the previous generation's Model 15 in photo 1. The Model 28 KSR with cover in place is shown below.





- Model 14 KSR: A Model 14 RO with a keyboard. Enables punching of paper tape off line. I've seen prices range from $15 to $75.

Teletype Model 15

This machine comes in two versions: a Model 15 KSR with page printer and keyboard and a Model 15 RO with page printer only (see photo 1). About 237,000 Model 15s were produced from 1930 to 1957, with Bell Systems receiving about 100,000 of these, the US government about 60,000, and Western Union, the wire services, newspapers, railroads, airlines and RCA Corp making up the balance. The page printer is usually set up for 60 words per minute, but occasionally 66 and 75 words per minute machines are encountered. The typical used price ranges from $30 to $200.

Teletype Model 19

A set of Baudot machines which includes a Model 15 page printer and keyboard, a paper tape perforator, a Model 14 TD reader, and a heavy duty loop supply. On some Model 19s, the punch is mechanically linked to the keyboard. Thus, tape must be punched locally and cannot be punched via computer control. The punch magnet requires a 100 VDC source capable of supplying one amp. The tape reader is usually connected in series with the keyboard so that serial Baudot signals can be generated from either unit. The surplus price ranges from $125 to $250.

Teletype Model 20

This model was used by the wire services and was similar to the Model 15 except it used a six level code for an expanded character set. This is a rare machine and I have not seen it advertised.

Teletype Model 26

Another rare, light duty machine used by the wire services.

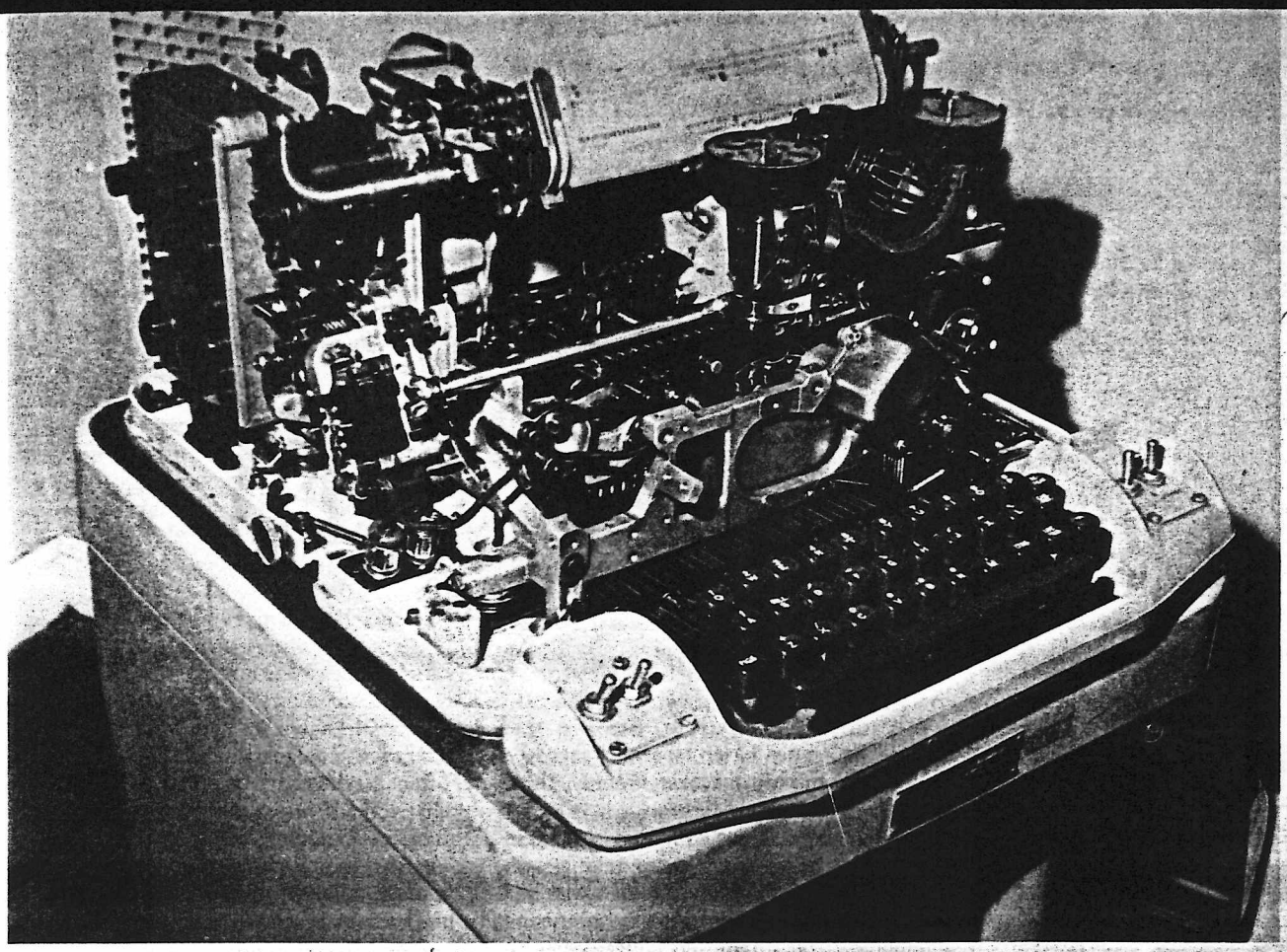


Photo 4: A view of a Kleinschmidt Model 150 KSR. This variant of the basic Teletype design (note its similarity to the Model 15) contains a current loop power supply.

Teletype Model 28

A general number for a series of Baudot machines, which includes ROs, KSRs, ASRs, TDs and reperforators. These are heavy duty machines like the Model 15, but generally operate at 100 words per minute. This is the most prevalent Teletype model used by the MARS people. Since production began in 1952, almost 300,000 Model 28s have been sold. This machine is still being produced so new replacement parts are plentiful. Photo 2 shows a Model 28 KSR.

There is an actual advantage of owning a used Model 28 as opposed to the popular Model 33 ASCII machines. This is because the Model 28 is a heavy duty machine made for 24 hours per day operation, year after year. However, the Model 33 is a light duty machine, which means it was designed for limited use: 3 to 4 hours per day, average. It will work for reasonably long periods of time but will require maintenance and adjustment more often than the Model 28 or any of the other heavy duty machines such as the Models 14, 15 and 19.

New prices from the Teletype Corp at last report were: Model 28 RO, $1750 to $2000; 28 KSR, $2000 to $2250; 28 ASR, $3300 to $3900; prices vary with options; delivery is from 6 to 12 months. None are in stock and each machine is built to order. Used prices range from $200 to $1500.

Teletype Model 29

Another rare machine, essentially a special Model 28 built for IBM.

Teletype Model 31

A rare teleprinter and keyboard combination, designed to print on 0.25 inch (0.64 cm) paper tape. This was a small piece of military aircraft equipment (nomenclature TT-30/AGA-1) which was designed shortly after World War II. It measured 9 by 14 by 12 inches (23 cm by 36 cm by 31 cm) and weighed 23 pounds (10.4 kg). Although assigned the Teletype number 31, there is little indication that there was ever any civilian production.

Teletype Model 32

Looks just like the ASCII Model 33 except it has a three row Baudot keyboard. This is a light duty, 100 words per minute Baudot machine still in commercial use. Available as RO, KSR and ASR. Used price

Continued on page 154

Continued from page 17

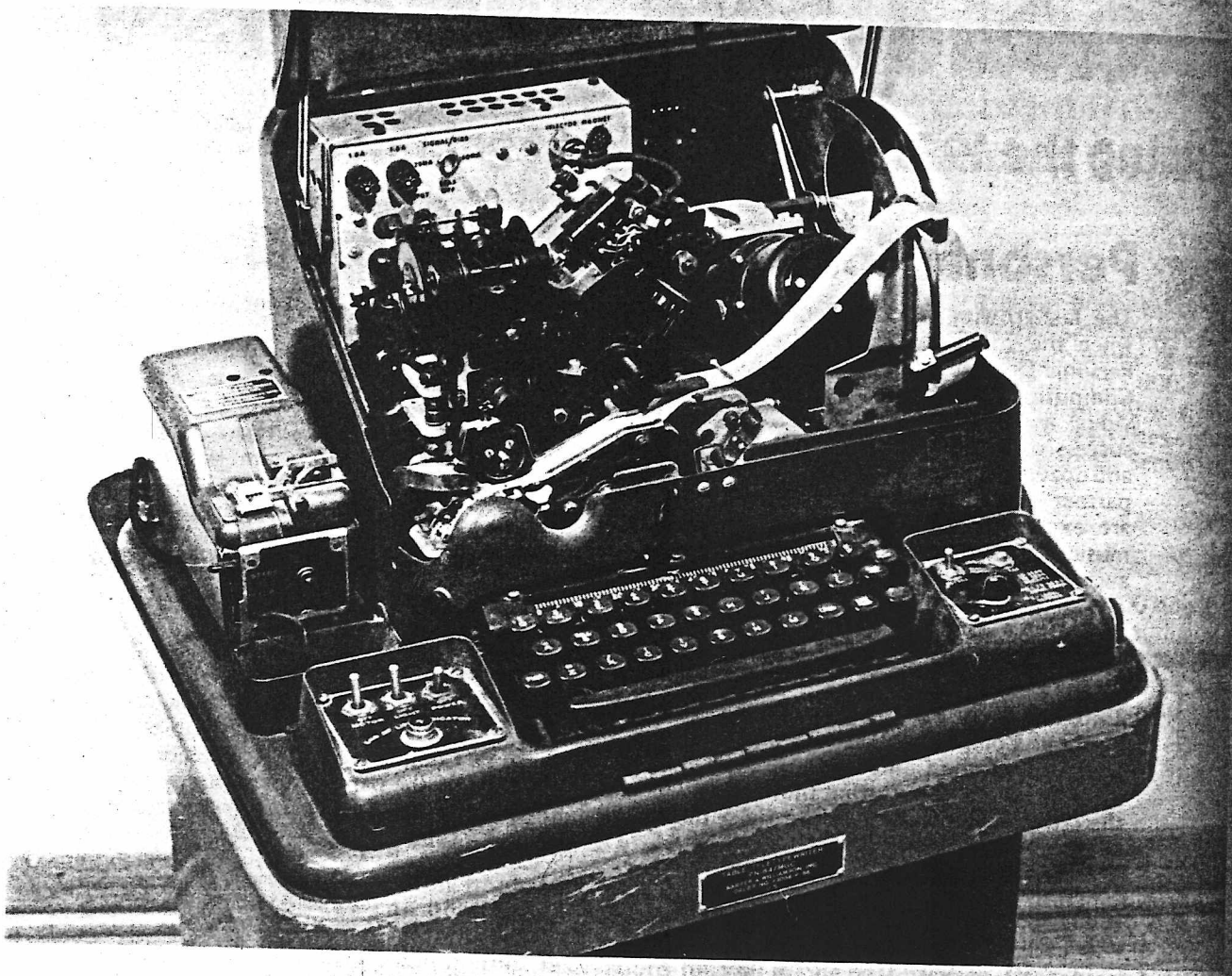


Photo 5: A view with cover lifted of the author's Kleinschmidt Model TT-76A (military nomenclature). This machine is a paper tape punch and reader only, which actually types characters on the tape. Like the Kleinschmidt in photo 4, this unit contains a loop power supply.

ranges from $100 to $1200, typically $400.

Teletype Model 33

This light duty eight level ASCII machine (see photo 3) is extremely popular in minicomputer circles, and could be called "the standard Teletype." It is widely sold with new computer systems. More than 600,000 Model 33s have been sold. New Teletype Corp prices are: 33 RO, $649; 33 KSR, $765; 33 ASR, $1090; a used ASR typically runs $700.

Teletype Model 35

A heavy duty version of the light duty ASCII Model 33 ASR. New price is $3800.

Teletype Model 37

A heavy duty, 150 words per minute ASCII machine with a full 255 graphic character set. Other features include software tab stop set, half line spacing, forward and reverse, etc.

Teletype Model 38

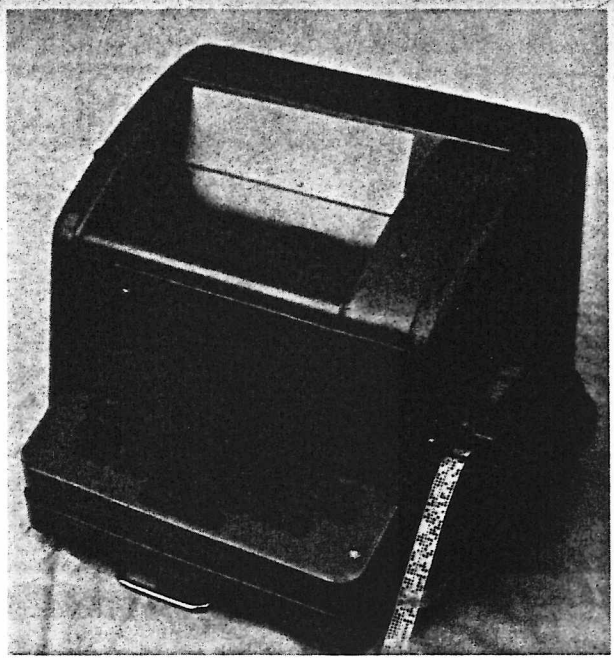
Another light duty machine, same as Model 33 but accepts standard computer forms. Has upper and lower case printing and two color ribbon capabilities.

Teletype Model 40

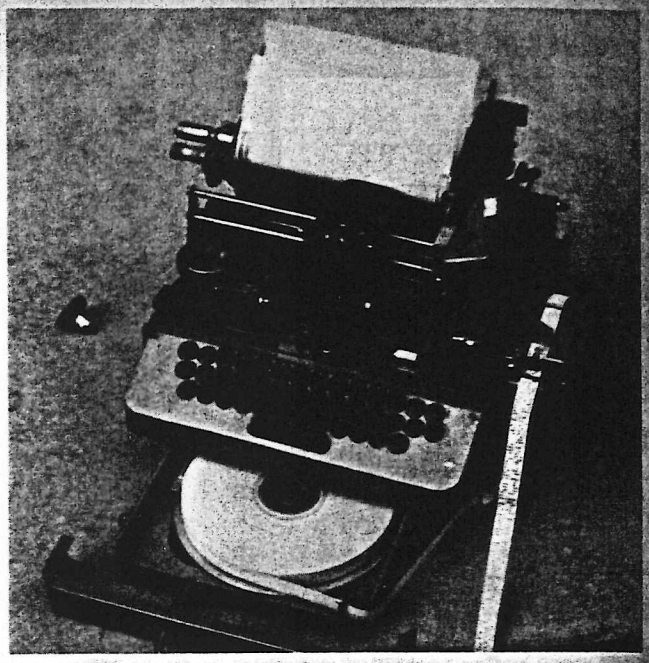
The latest ASCII Teletype Corp machine includes a line printer, keyboard, CRT display, etc. The high speed line printer with this set is an 80 or 132 column, 300 lines per minute, heavy duty unit. Prices in the $1400 to $2000 range.

Kleinschmidt

Kleinschmidt machines were manufactured during the 1930s, 1940s, and 1950s in direct competition with the Teletype Corp, and more recently by the Smith Corona Marchant Corp of Dearborn MI. This line of devices includes the following: page printer and keyboards (TT-117), paper tape reader



(a)



(b)

Photo 6: Two views of the Creed 75 unit. In (a), the unit is shown with the cover in place as it might be seen in use. In the second view (b) the cover has been removed, and the paper tape supply drawer opened for inspection.

and punch sets (TT-179) or a set designated the TT-100, which includes a page printer, keyboard, table and power supply. These units are fairly common although not in as plentiful supply as their Teletype counterparts. As with Teletype Corp, many other Kleinschmidt configuations were manufactured for the military under various nomenclatures such as TT-65, TT-98, TT-271, etc. For example, a Model TT-24 is a 100 word per minute KSR machine. Photo 4 shows a Kleinschmidt Model 150 KSR, which is typical of this line. My personal typing paper tape reader and punch set is labeled "TT-76A/GGC" and is shown in photo 5. Price range, $20 to $150.

Lorenz

A German company which, like Kleinschmidt, has manufactured Model 15 KSRs and 15 ASRs. These machines are relatively rare in the US.

Mite

This New Haven CT company has manufactured small Baudot page printer and keyboard machines for use in submarines and by the US Marine Corps. This device is relatively rare on the surplus market.

Creed Model 75 Teletypewriter

ITT Creed is the English counterpart of Teletype Corp. This company manufactured several hundred special Creed Model 75 teletypewriters for Spiegel in Chicago in the early 1960s, which were eventually passed on to a company named Wilcox Enterprises in Naperville IL.

The basic unit shown in photo 6 consists of a page printer, keyboard and paper tape punch, but no paper tape reader. This machine uses a Baudot code at 10 characters per second. Some differences from American machines are that the keyboard and printer are mechanically linked, ie: no loop supply is needed for local, typewriter like operation. The printer is run by activating a 24 VDC solenoid for space and another one for mark; thus there is no current loop.

The keyboard output consists of five parallel bits and one strobe bit to indicate character ready, although it can be rewired for serial operation.

Also available from Wilcox Enterprises are interface kits, programs, and erasable, programmable, read only memories which adapt the machine for use with 8008 and 8080 systems.

What Characteristics Should I Look For?

Once you've found a Baudot machine and are thinking about whether or not to buy it, two items are of key importance besides general mechanical condition; these are machine speed and character types.

Basically these units were supplied with three different kinds of character sets: com-

SUPPLIERS OF BAUDOT EQUIPMENT

Among the companies which advertise teleprinters in the various amateur radio magazines, computer journals and other "surplus" advertising spots are the following:

Goodman, 5454 S Shore Dr, Chicago IL 60615, (312) 752-1000.

Teletypewriter Communications Specialists (Van's Electronics), 550 Springfield Av, Berkely Heights NJ 07922, (201) 464-5310.

Andy Electronics Co Inc, 6319 Long Dr, Houston TX 77087, (713) 641-0576.

Typetronics, POB 8873, Ft Lauderdale FL 33310, (305) 583-1340.

Lawrence R Pfleger, 532 W Wilson St #1, Madison WI 53703, (608) 255-2935.

Atlantic Surplus Sales, 3730 Nautilus Av, Brooklyn NY 11224, (212) 372-0349.

Henry Holder, (817) 429-3021.

Don C Dando, Rt 4, POB 454, Liberty MO 64068.

BVE Enterprises, POB 73, Paramus NJ 07652.

Alltronics-Howard Co, POB 19, Boston MA 02101, (617) 742-0048

Wilcox Enterprises, 25W178 39th St, Naperville IL 60540, (312) 420-8601

This list was culled from the publications circa July 1976. No inventory or prices are listed since as one supplier (Goodman) put it in a conversation, "I have hundreds of 15s and 19s now, but they sometimes move slow, or sometimes fast, and I could sell 'em all tomorrow." In each case, where available, a phone number is listed so that the latest availability can be checked.

mercial, weather and stock market. Table 1 lists the characters which correspond to particular 5 bit Baudot inputs. Note that in all three cases, the letters, lower case, are the same, but the figures, upper case, are different; in fact some of them are downright strange!

Obviously the most desirable character set for microprocessor use is the commercial or communications group, but don't let a stock market or weirdo weather font stop you! Replacement commercial keys are available for $0.50 to $1 each from most of the sources listed.

There are four printing speeds associated with these old machines, those being 60, 66, 75 and 100 words per minute. Table 2 summarizes the data rates, bit pulse widths, character speeds, etc, for each of the four speeds. More about how to increase machine speeds later.■

Next Month: *Baudot teleprinter information continues with a discussion of interfacing techniques and problems.*

Byte vol 2 #5
May 1977

Last month author Mike McNatt described a number of Baudot type teleprinters that are available on the surplus market today. Let's assume that you have since rushed out in the intervening period and acquired your own Baudot teleprinter. To use it with your system, some method must be devised to enable communication between the processor and the serial Baudot device, Mike's topic in this second part of a 3 part series.

A Guide

Michael S McNatt
4658 E 57 St
Tulsa OK 74135

To use a Baudot page printer or keyboard with your computer system, some method must be devised to enable communications between the processor and the serial Baudot device. Traditionally, these interface methods have been classified as either software or hardware. Both methods seem to have supporters who are radically convinced that the method they've chosen is the best. Strictly speaking, none of the Baudot machine interface methods can be called purely software or purely hardware; they all lie somewhere in between. However, the hardware code converter obviously requires the least amount of new software generation.

Hardware Interfaces, Code Converters

The chief advantage of these converters is that they allow the use of Baudot machines with computer software which is written to input and output exclusively in ASCII code. This is a major factor when you consider that much of the microcomputer software available is written with ASCII IO. Figure 1 illustrates the use of these converters, which consist of circuitry to (a) convert the parallel ASCII code from the processor data bus into serial Baudot code for use with page printers or paper tape perforators or (b) convert the serial Baudot output from a keyboard or tape reader into parallel ASCII code for presentation to the processor data bus.

When using hardware conversion from ASCII to Baudot, a handshaking arrangement is used whereby the Baudot machine tells the processor when it is ready to accept a character. Parallel ASCII data generated by the program's output routine is then strobed into the converter via a load pulse from the processor.

The converter's ready line changes logic levels, indicating that it is in a busy state. During this state, the processor stands by while the converter

1. generates a figures or letters shift character if required,
2. converts the shift character into serial pulses which are sent to the Baudot machine,
3. converts the parallel ASCII character just loaded into an equivalent parallel Baudot character via a programmable read only memory lookup table, and
4. converts this new Baudot character into serial pulses which are sent to the Baudot machine.

The converter's ready line then changes back to its original state, indicating that it is ready to accept another character.

A variation of this approach which may not require busy and ready handshaking is to use a long serial shift register buffer. This would in effect compensate for speed dif-

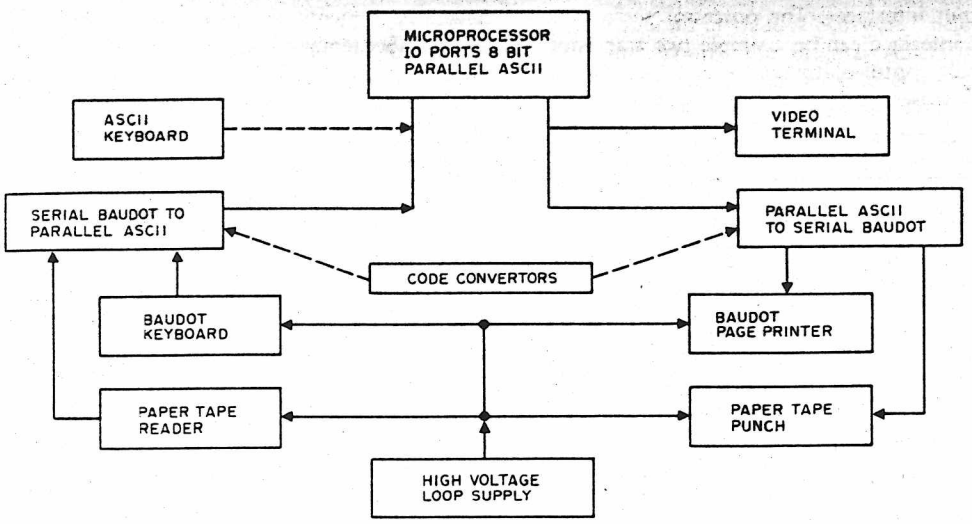


Figure 1: The block diagram describing the use of hardware code converters interfaced with a microprocessor.

to Baudot Machines:

Part 2, Interfacing Techniques

ferences in an ASCII 110 bps serial output and the slower 45 to 75 bps rates of the Baudot machines. Since most interface messages are less than one line of 88 characters in length, one line of shift register buffering may suffice; see reference 22, volume 2, #1, January 1976, page 9.

When using hardware conversion from Baudot to ASCII, serial pulses from the Baudot machine are loaded into a shift register and presented as five parallel lines representing one character to another read only memory lookup table. Circuitry in the converter keeps track of whether the character has been preceded by a letters or figures code, and presents this information to the read only memory as a sixth input line. When the parallel ASCII output from this memory is ready, the converter generates a data ready pulse, during which the processor loads in the ASCII character. This conversion operates in a manner similar to that of a standard ASCII keyboard which generates a strobe or ready pulse after each keystroke, indicating that the character may be loaded into the processor.

Circuits for accomplishing these conversions have been published, (references 11, 11a, 15, 16, and 21) and are available as printed circuit board assemblies from several of the companies listed in a separate box labelled "Sources of Baudot Hardware." References 9 and 10 present parallel to parallel code conversion circuits which require an additional UART or similar circuitry to obtain serial data for, or from, the Baudot machine.

Software Interfaces

In the purists' software approach, one routine is used to construct the serial Baudot code for output, and another routine is used to decipher the incoming Baudot code for input to the operating program. Routines used to accomplish these functions are sometimes referred to as timed software, since the bit timing of each character, 60 wpm, 100 wpm, etc, is derived from software timing loops. For example, to derive the precise 22.0 ms pulse periods required for 60 words per minute operation, a set of instructions which requires 12.7 μs to execute would be repeated 1,732 times for each bit of serial data.

The hardware requirement is held to an absolute minimum in both the input and

output interfaces. The processor serial output interface can be a simple two transistor circuit, optoisolator or relay which opens and closes the teleprinter's current loop.

Several comments observed in the *RTTY Journal* (September 1973, page 13, for example) indicate that a Motorola MOC 1000 optical coupler has been used successfully up

GLOSSARY OF TELETYPE TERMS

AFSK: Audio Frequency Shift Keying; see Frequency Shift Keyer.

ASR: Automatic Send and Receive. Refers to a teleprinter system which includes a page printer, a keyboard, a paper tape punch and a paper tape reader.

Chad, Chadless: A classification of paper tape punch machines, referring to whether the chad or small paper circles remain as a part of (chadless) or are completely removed from the punched tape. A quote: "The chadless tape does not wind well, takes up more space when wound and occasionally will not read correctly if the little tabs of paper get stuck back in the holes." (Dan Nolan, *RTTY Journal*, February 1976, page 17.) However, these potential problems may be insignificant if a particular chadless machine is priced right.

Clutch: A term used to denote that current is continuously flowing in the Teletype loop supply, uninterrupted by any data input. The clutch or marking state is used to prevent the machine from running open and producing nothing but mechanical clatter.

Converter, Terminal Unit (TU), or Demodulator: A radioteletype (RTTY) term which indicates a device that accepts the two audio tones, representing mark and space, from the output of a receiver and converts them into current loop pulses for displaying characters on a page printer.

CR and LF: Carriage Return and Line Feed.

Demodulator: See Converter.

Frequency Shift Keyer, FSK: A radio Teletype term which indicates a device that converts the current loop pulses from a keyboard or paper tape reader into audio tones. An amateur radio transmitter then converts these tones to radio waves suitable for communications use.

Governor Motor: A nonsynchronous teleprinter motor, the speed of which is normally adjusted mechanically by viewing white strobe dots on the rotating assembly through a slit shutter tuning fork. Some hams have built speed controls for these machines, enabling electrical adjustment from 60 to over 100 words per minute.

KSR: Keyboard Send and Receive. Refers to a teleprinter system which includes a page printer and a keyboard only.

Ham: A radio amateur.

Loop Supply: A DC power supply used to drive teleprinter selector magnets; capable of supplying 20 or more commonly 60 mA at 130 to 260 VDC. Below 130 V, loop time constants can cause difficulty. Western Union telegraph loops usually work with 260 V.

Mark or "Closed Loop": A condition of steady current flow in the teleprinter loop; see clutch.

MARS: The Military Affiliate Radio System, whereby civilian amateur radio operators, hams, assist the armed forces with message communications, at the same time acquiring valuable training in military communications procedures. MARS is operated jointly by the Navy, Air Force and Army.

Paper Tapes: The narrow 0.375 in (0.953 cm) tape is usually gum backed and is used with a strip printer which just types messages on narrow tape. Western Union attaches strips of this tape to paper in their telegram operations. The wide 0.6875 inch (1.746 cm) tape is used with typing reperforators, chadless, and nontyping reperforators, chad type, which punch holes in the tape.

Perforator: A unit which can punch tape from an attached Baudot keyboard input only because the perforator operates from mechanical linkages to this keyboard. Cannot punch tapes from an external current interrupting source.

Reperforator: Derived from *re*ceiving *perforator*. This unit can receive serial Baudot paperpunching messages from both the keyboard and an external current loop interrupter.

RO: Read Only. Refers to a teleprinter system which includes a page printer only.

RTTY: Radioteletype: A form of amateur radio communications which uses teleprinter machines for transmitting and receiving messages via radio waves.

Space or Open Loop: A condition of no current flow in the teleprinter loop.

Stunt Box: A mechanical teleprinter attachment which performs certain nonprinting functions or stunts. These may include turning on a light, switching from nonprint to print, automatic carriage return and line feed, vertical and horizontal tabulation, backspace, two color ribbon, etc. These are rather complicated, Rube Goldberg contraptions, and are difficult to service.

Synchronous Motor: A motor which rotates synchronously at a constant speed which is directly proportional to the AC line frequency.

Terminal Unit, TU: See Converter.

Transmitter Distributor, TD, or Tape Reader: A machine which senses the characters punched into a paper tape and sends them automatically at a constant speed over a teleprinter loop circuit.

Typing Reperforator or Tape Punch: A machine which punches Baudot coded chadless holes into a paper tape and prints the associated characters right on the tape.

WPM: Words Per Minute.

UNIVERSITY OF CHICAGO LIBRARY

to 130 VDC in this application. The serial input interface is an equally simple minimum component count circuit (see references 13 and 14).

There are two ways to input and output the serial Baudot data with this software approach:

1. Use operating software which inputs and outputs Baudot directly, as in the Baudot Monitor and Editor programs listed in reference 20.
2. Use operating software which normally uses ASCII input and output. However, provide additional code conversion lookup table software (see "Sources of Baudot Software" below) and modify the original ASCII programs to call these code conversion routines. Therefore, serial Baudot is input and output at the processor IO port. Obviously, if the programs are in read only memory and not programmable read only memory, they cannot be modified to take vectors to the software conversion routines.

Both approaches require the use of the timed software described earlier.

An In Between Interface

Another processor to Baudot machine interface method might be called a hybrid ap-

SOURCES OF BAUDOT SOFTWARE

1. MITS, 2540 Alamo SE, Albuquerque NM 87106. Selected programs from the Altair Software Library:
   (a) =728752 by J R Scott, $2, 36 byte program translates some Baudot character codes to ASCII equivalent.
   (b) =5-24-762 by Walter King, $2, 50 byte program loads object code from 5 level paper tape.
   (c) =5-24-765 by Walter King, $2, 71 byte program punches 5 level object tapes which can be read by #5-24-762.
   (d) =6-18-761 by James B Hansen, 215 byte program, title: ASCII to Baudot Translate Routine.
   (e) =6-22-761 by Jim Wiggins, (1) 106 byte program, title: Tape Load – Octal TLQ, (2) 186 byte program, title: Tape Dump – Octal TDQ, for Baudot Teletypes.

   The prices shown assume membership, $10 per year, in the Altair User's Club. An interesting note is that MITS is accepting no more Baudot code converter or Baudot paper tape punch and load programs into their library because it is overstocked with these types of programs.
2. M6800 User's Group Library, Motorola Inc, Microsystems, 3102 N 56th St, Phoenix AZ 85018. Reentrant ASCII to Baudot Converter Subroutine, Astrobaud. Other Baudot related programs may become available. A two year membership into this user's group is available for $100 or to anyone who submits an entry to the Program Library.
3. Intel Corp, User's Library, Microcomputer Systems, 3065 Bowers Av, Santa Clara CA 95051. Paper Tape Conversion program in 4040 language, converts 5 level paper tape information to ASCII 8 level paper tape. Other 4040 and 8080 Baudot code related programs may become available. A one year membership in the user's group is $100 or is available if a program is submitted to the library.
4. The Digital Group Inc, POB 6528, Denver CO 80206. Software Package HAM-1, Amateur Radio Ham Cassette for 8080, includes programs to send and receive RTTY Baudot, at 60, 66 and 100 wpm; $5 for cassette plus documentation.
5. See references 22a, 22b, and 23 of the bibliography.

SOURCES OF BAUDOT HARDWARE

1. Processor Technology, 2465 Fourth St, Berkeley CA 94710. 3P + S IO Module, $125 kit, $165 assembled, Altair/IMSAI plug compatible. Includes serial IO ports under software control, from 35 to 9600 bps for use with older Baudot machines.
2. HAL Communications Corp, 807 E Green St, POB 365, Urbana IL 61801. MCEM-8080 Microcomputer System, $375 assembled. Software monitor supports ASCII or Baudot code; optoisolator loop current driver and receiver.
3. Mini Micro Mart, 1618 James St, Syracuse NY 13203. Code Converter Boards: (1) Parallel ASCII to Serial Baudot kit #GPA-B, $34.95, (2) Serial Baudot to Parallel ASCII kit #TVT 2540 – BSI, $34.95.
4. MITS, 2450 Alamo SE, Albuquerque NM 87106. The Altair 680 microcomputer, $466 kit, is listed as having a five level Baudot interface option, an additional $42.
5. IMSAI, 14860 Wicks Blvd, San Leandro CA 94577. Two of the IO boards available from this company offer Baudot machine compatible serial output bps rates.
   (1) SIO 2-1, $125 kit, 75 bps minimum, compatible with 100 wmp machine such as Model 28s and property geared Kleinschmidts.
   (2) MIO, $195 kit, continuously selectable from 40 to 9600 asynchronous, compatible with 60 wpm, 45 bps, Model 15s as well as all other speeds.
6. Southwest Technical Products Co, 219 W Rhapsody, San Antonio TX 78216.
   (1) ASCII to Baudot Converter kit, $24.50, PC board, $4.35. Circuit presented in reference 15.
   (2) Baudot to ASCII Converter kit, $19.50, PC board, $4.35. Circuit presented in reference 16.
7. Polymorphic Systems, 737 S Kellogg, Goleta CA 93017. A Serial IO Option, $75 kit, can be added to the Poly-88 microcomputer kit which provides a 50 bps minimum IO rate.
8. Ohio Scientific Instruments, OSI, POB 374, Hudson OH 44236. Model 412B Supercard is designated for Baudot Terminals, $129 kit, based on MOS Technology 6502 microprocessor.
9. See reference 17.

proach, whereby parallel Baudot characters are generated by the software. These characters are presented to the data bus where busy and ready handshaking logic described in the above hardware interface description is used. In effect, the software performs the code conversion or parallel IO function and the hardware performs the parallel to serial conversion and character bit timing functions.

A UART or similar circuit (see reference 17) can be used to implement these functions. The whole procedure operates in reverse when the serial Baudot characters are input to the hardware portion of the interface and presented in parallel form to the data bus.

The Interrupt Driven Interface

These old surplus Baudot machines may of course be interfaced to the processor using an interrupt scheme, with priorities assigned to each Baudot device along with other system peripherals. Interrupt schemes are beyond the scope of this discussion since there are so many variations of software and hardware interrupt techniques, and they vary according to the processor being used. Many, and possibly most, computer hobbyist systems do not use interrupts at all. Usually in this class of system, either there are few peripherals being used or the operator has lots of time on his hands to WAIT on the peripherals, or both. One example would be a system with a keyboard and page printer combination operating with monitor and debug programs.

For a system with several IO devices and very busy software, setting up an interrupt scheme may become a matter of necessity. If

REFERENCES ON BAUDOT TELETYPES

Many of these references, especially the amateur radio magazines and books, may be available for loan from a friendly ham in your area. Some ham operators have been known to hoard all of the issues of some of their type journals, such as *Ham Radio, CQ, 73,* or *QST* magazines. It should be known that a similar affliction has been found with many BYTE readers . . .

1. *RTTY Journal*, POB 837, Royal Oak MI 48068, 10 issues per year, $.35 per issue.

2. *RTTY from A to Z* by Durward J Tucker, Corvan Publishing Corp, 1970, 214 pages, $5. Detailed information on operation, mechanical characteristics and maintenance of practically all of the older Teletype machines.

3. *The Teleprinter Handbook*, first edition by D J Goacher and J G Denny, Radio Society of Great Britain, 1973, $14.95. Distributed in US by Ham Radio Books, Greenville NH 03048. An engineering textbook covering the theory and practice of radio teleprinter equipment; 30 pages of pictures and wiring diagrams of Models 15, 19, and a few pages on the Model 28. Also covered are the design of power supplies, demodulators and Teletype test equipment.

4. *Specialized Communications Techniques for the Radio Amateur*, by the American Radio Relay League, 1975, 208 pages, $3. Chapter 5: Radioteletype, 70 pages. A condensed overview of many of the Baudot machines, a table of Teletype Gear numbers for Models 14, 15 and 28, and many other basics.

5. *The New RTTY Handbook* by Byron H Kretzman, Cowan Publishing Corp, 1962, 191 pages, $3.95. Includes a brief history of Baudot teleprinters and some equipment descriptions.

6. *RTTY Handbook* by Wayne Green, Tab Books, 1972, 320 pages, $6.95. Many basic amateur radio operating circuits such as converters, exciters and FSK circuits.

7. Shinsel, Vic: "RTTY Signal Generator," *Ham Radio Magazine,* March 1971, pages 23 to 29. RTL version of RY generator circuit.

8. Dean, J A and Rupley, J P: RCA Application Note ICAN-6267, "Astable and Monostable Oscillators Using RCA COS/MOS Digital Integrated Circuits," 8 pages.

9. *The TTL Cookbook* by Donald M Lancaster, Howard Sams Inc, pages 152 to 153. A parallel Baudot to parallel ASCII conversion circuit is presented which uses 5 ICs including 2 PROMs.

10. Ellsworth, Cole: "ASCII to Baudot Converter," *73 Magazine*, February 1976, pages 52 to 57. A 15 IC parallel ASCII to parallel Baudot code converter. Additional UART circuit required to obtain serial Baudot for teleprinters.

11. Hutton, Louis I: "Build This Exciting New TVT," *73 Magazine*, March 1976, pages 76 to 83. Included is a Teletype current loop Baudot code to parallel ASCII converter for driving a video terminal display, page 80.

11a. Hutton, Louis I: "Code Converter Using PROMs," *73 Magazine*, May 1976, pages 84 to 86. Improved circuit using readily available PROMs; refers to reference 11.

12. Andreasen, C Warren: "A Versatile TTY Generator," *73 Magazine*, April 1976, pages 90 to 91. Circuit is given that counts from 00000 to 11111 and converts to serial Baudot, thus generating a pattern of all Baudot letters and figures.

13. Whipple, Dick, and Arnold, John: "A Very Cheap IO — the Model 15," *73 Magazine*, May 1976, pages 77 to 80. Presents both software and hardware interfaces.

14. King, Walter S: "Interfacing the 60 mA Current Loop," *BYTE* magazine, August 1976, pages 96 to 97.

UNIVERSITY OF CHICAGO LIBRARY

this is the case, information on interrupt interfacing techniques and hardware may be obtained from a variety of references, including microprocessor manufacturers' application literature or any of the excellent programming books or publications available (references 23b and 26) which deal with this subject.

Useful Miscellaneous Items

Cleaning an Old Teletype: In the course of researching for this article, I ran across an employee of a national teleprinter refurbishing house for Model 28s, 33s, and 35s, who offered some interesting suggestions for cleaning a Model 15. He said that this procedure could be followed if one obtained an extremely dirty teleprinter "fresh" out of a damp basement. The steps are as follows:

1. Thoroughly clean the machine with hot water. An ideal place for this is at a self-service car wash facility.
2. Force air dry the machine with an air gun from, for instance, one available at the auto service station with the car wash.
3. Spray all parts with a light lubricant such as WD-40.
4. Lubricate as required with oil and grease. Details of these procedures are contained in references 1 through 6. Pay particular attention to lubricating the felt separating pads, which should be saturated.

One thing I might add is to clean all electrical contacts, especially the keyboard, after the above sequence. A standard pencil eraser rubbed over the contacts until clean is an excellent way to accomplish this.

15. Smith, Roger L: "ASCII to Baudot," *Radio Electronics*, March 1976, pages 51 to 58. Corrections, April 1976, page 58, and August 1976, pages 16 and 22. Hardware code converter.

16. Smith, Roger L: "Baudot to ASCII," *Radio Electronics*, April 1976 pages 57 to 59. Hardware code converter.

17. Pasco, Richard C: "Converter Lets Processor Drive Teletypewriter," *ELECTRONICS*, October 30 1975, pages 97 to 98. Simple 6 IC TTL circuit which accepts 5 or 8 bit parallel data and converts it to serial current loop pulses at any desired bps rate.

18. *Reference Data for Radio Engineers*, fifth edition, pages 35 to 40, table 24: "Moore ARQ Code (Compared with 5 Unit Teleprinter Code)" and pages 35 to 38, table 22: "Timing of Five Unit Start Stop Teleprinter Codes."

19. Whipple, Dick, and Arnold, John: "Inexpensive Paper Tape Subsystem: Using 5 Level Tape with Computers," *73 Magazine*, July 1976, pages 145 to 148. Includes listings of paper tape read and punch software for Teletype Models 14 and 19.

20. Whipple, Dick, and Arnold, John: "A Baudot Monitor/Editor System . . . program listing for the 8080," *73 Magazine*, August 1976, pages 102 to 105. Uses the software IO described in reference 19. Also uses split octal addresses and octal data notations.

21. Guthrie, R David: "ASCII/Baudot with a PROM," *73 Magazine*, June 1976, pages 114 to 119. Article title is a misnomer – actually a serial Baudot to parallel, or serial, ASCII hardware code converter circuit is presented for any desired data rate.

22. *The Computer Hobbyist Group (North Texas) Newsletter*, 2377 Dalworth 157, Grand Prairie TX 75050.
(a) Volume 2, #4, April 1976, pages 7 to 13. Contains listings of ASCII to Baudot and Baudot to ASCII conversion software for both the 8080 and the 6800 by Tom Gentry, James Brown, and others on the newsletter's "conversion committee."
(b) Volume 2, #5, May 1976, pages 7 to 11. Contains a listing of ASCII to Baudot conversion software for the F8 microprocessor by Harold A Mauch.
(c) Volume 2, #2, February 1976, "TTY Interfaces" by Harold A Mauch, pages 13 to 15.

23. *The Computer Hobbyist*, POB 295, Cary NC 27511. Back issues are $.65 each.
(a) Volume 1, #3, January 1975, pages 19 to 21, "Interfacing a 5 Level Teleprinter" by Steve Stallings, software courtesy of Richard Smith. Contains listings of an ASCII to Baudot Translate routine, a Teletype Demo Program which generates lines of ripple pattern on the page printer, and a Baudot Teletype Output Routine which accepts Baudot characters and enables outputting in Baudot directly, if called as a subroutine, rather than converting from ASCII to Baudot.
(b) Volume 1, #9, February 1976, pages 11 to 13, "Interfacing the Altair 8800, Part 2" by Hal Chamberlin. Includes an excellent discussion of simple single device hardware and software interrupts, vectored and polled interrupts, and an example of a keyboard interface using the interrupt technique.

24. Teletype Corp, 5555 Touhy Av, Skokie IL 60076.

25. Nettles, Walter: "Instructions for Hand Operated Typewriter Ribbon Reinking Device," reference 1, Volume 12, #6, page 16.

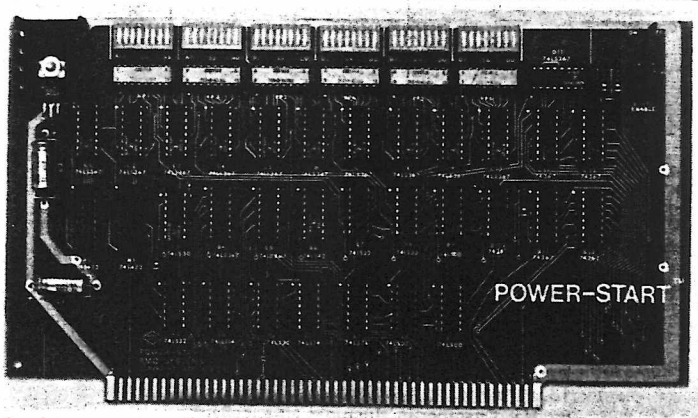
26. *How to Use the Nova Minicomputers*, by Data General Corp, 1971, Section 2.4 "Program Interrupt." Contains a discussion of interrupt servicing and priority and programming suggestions such as when to use interrupt.

Ribbon Renewing: When the black print ribbon begins to fade or dry up, additional ink may be applied on the top and bottom of the ribbon to provide many more miles of heavy duty use from a single roll of ribbon. Also, the printing is much darker and easier to read on both page printers and typing tape reperforators. Inks suitable for this purpose include stencil ink, stamp pad ink thinned with wood alcohol (methanol), and mimeograph or duplicator ink thinned with either naptha or turpentine; see reference 25. One ham I spoke with bragged of using 3-in-1 household oil as a thinning agent. Regardless of which type of ink is used, apply new ink sparingly until you get the knack of reinking. Allow at least 24 hours for the ink to soak into the ribbon and partially dry out before using on a printing machine.

Increasing the Operating Speed of Baudot Machines: There have been many comments in the hobbyist literature regarding the effects of increasing the speed of the older Baudot machines. Of the more popular units such as Models 14, 15, 19 and 28, only the Model 28 and some Model 14s were actually designed to operate at 100 words per minute. The other models were designed for slow speed operation and simply do not hold up well when run faster. Gordon White, the "Surplus Sidelights" columnist for *CQ* magazine, had these comments in *CQ*, August 1971, page 87:

> "It is possible to gear the old equipment up to turn 100 wpm, but particularly the Model 15 and 19 page printers do not last long at 100 speed. The Model 14 typing reperforator will do a little better, but still hops around when forced to run at 100 speed. The Model 14 transmitter distributor appears to serve adequately at 100 wpm, as does the Model 14 nontyping reperforator, if kept well oiled.
> "The older machines will run at 67 or 75 wpm with a gear change . . ."

The gear changes referred to are relatively simple to achieve. Faster gear sets may be obtained from most of the suppliers of Baudot teleprinters. Also, a full page table of gear set numbers required for various speeds on Model 14s, 15s, and 28s is shown on page 111 of reference 4.■



- Never Toggle in a Bootstrap Again
- Never Flip Sense Switches Again
- Run Without a Front Panel
- Plug Into Backplane
- Use With Your Own ROM
- Completely Configurable

The Automatic Boot Loader

POWER-START™

RMQ Systems' new Altair 8800-compatible POWER-START™ board is supplied with complete documentation and configuration information. Components are of the highest commercial quality, including double-sided printed circuit board with solder mask on both sides and silk screened legend, low-profile gold-inlay IC sockets, and gold-plated bus contacts.

POWER-START™ Kit (PS-2K) 165.00 Kit (PS-1K) with ROM 195.00
Assembled (PS-2A) 265.00 Assembled (PS-1A) with ROM 295.00

To Order or for More Information:

RMQ Systems
Research & Development Division of
computer kits, inc.
the systems store
1044 University Ave., Berkeley, CA 94710 (415) 845-5300

Interface Age
May 1978
Vol 3 no 5

INSIDE ASCII

By R. W. Bemer

The data alphabet called ASCII (Figure 1, page 98, and Reference 1), also has two other names—International Standard 646 (the ISO Code [Reference 2]) and Alphabet No. 5 of CCITT (the International Consultative Committee for Telephone and Telegraph). It is used throughout the world, incorporated in billions of dollars of equipment.

But is it used correctly and wisely? Not always. There are misinterpretations, and gaps in definition that permit nonstandard usage. This article (in three parts) will give you the background, peculiarities, preferred practices, and new developments for ASCII. You will find a lot of information not too generally known or realized; it should help in the correct and safe usage of ASCII. For additional help, you can reference the various national and international standards given in Table 1. Some other detailed articles are listed in References 3, 4 and 5.

	ISO	ECMA	ANSI	FIPS PUB	CSA	BS	AS	CCITT	JIS	GOST
Binary-coded Character Set	646	6	X3.4-1977 $4.50	1	Z243.4	4730	1776	V.3	C6220	13052-67
Graphics for Control Characters	2047	17	X3.32-1973 $3.50	36		4730				
Character Set for Handprinting	97/3 N119		X3.45-1974 $5.75	33	Z243.34.1					
Additional Controls Character Imaging		48	BSR X3.64							
4-bit Sets	963	14		15	Z243.6	4731/1	1070			
Code Extension Techniques	2022	35	X3.41-1974 $6.00	35	Z243.35	4953				
Registration Procedures for Escape Sequences	2375									
8-bit Coded Character Set	DIS 4873	43	X3L2/77/08							
Character Set for 7 x 9 Matrix Printers		42								
Keyboard	2530	23	X4.14-1971 $3.75			4822/1	1922			
Character Sets for Programming Languages	97/5 N436	53								

Legend

ISO - International Standards Organization
ECMA - European Computer Manufacturers Association
ANSI - American National Standards Institute
FIPS - Federal Information Processing Standard
CSA - Canadian Standards Association
BS - British Standard
AS - Australian Standard
CCITT - Consultative Committee International, Telephone & Telegraph
JIS - Japanese Industrial Standard
GOST - USSR Standard

Table 1.

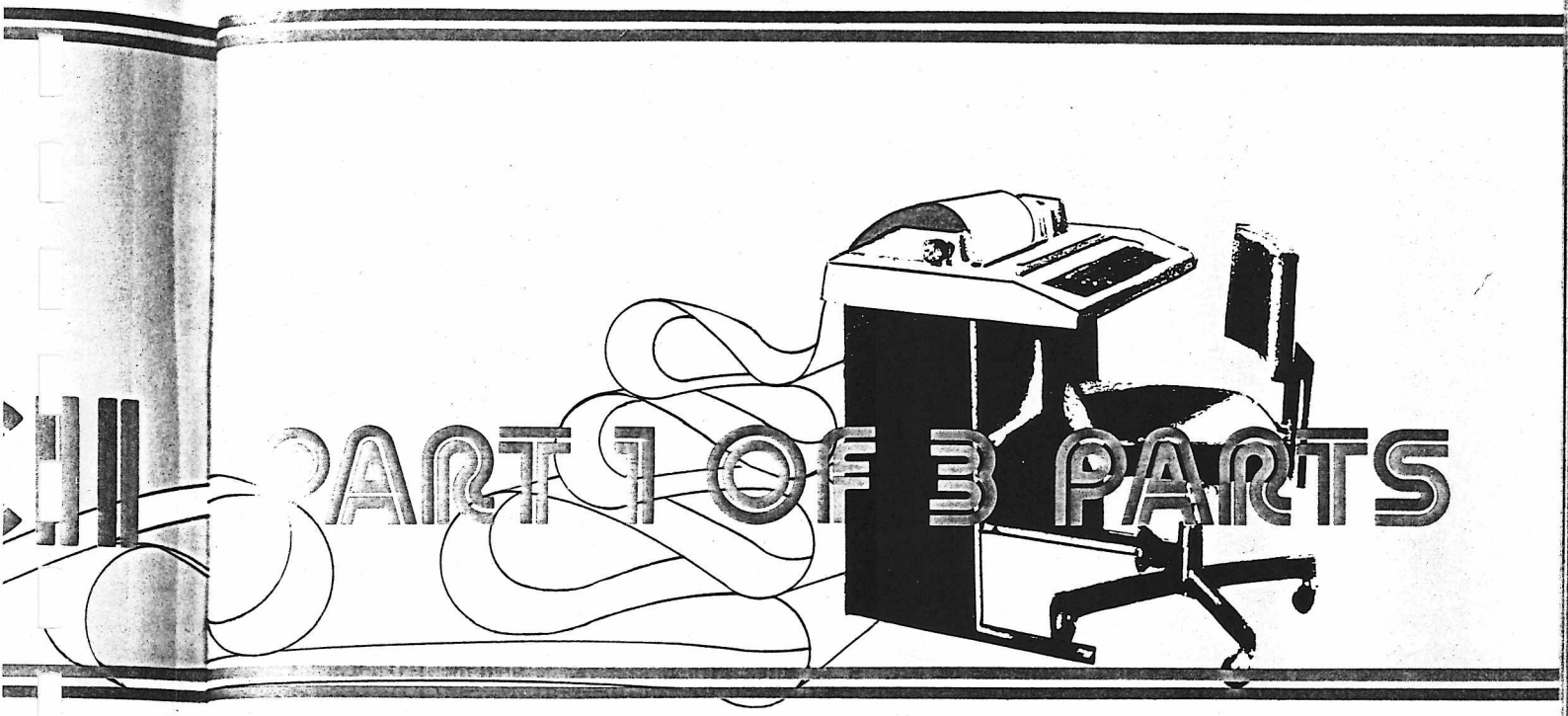
STICKS 4-7

ASCII, as a 7-bit code, is usually represented in 8 columns of 16 positions. The row positions are 0000 through 1111, the low-order 4 bits, 0 through 15 in decimal. The columns are 000 through 111, the next higher 3 bits, 0 through 7 in decimal. For some reason, the developers of ASCII found it convenient to refer to these eight columns as "sticks." So shall we. Each position will be represented in this article by its usual decimal representation. For example, capital A is position 4/1. Figure 2 is a representation of ASCII that is more convenient to those working in octal, rather than hexadecimal, notation.

HIGH ORDER OCTAL DIGITS →	00	02	04	06	10	12	14	16	← LOW ORDER OCTAL DIGIT
	NUL	DLE	SP	0	@	P	`	p	0
	SOH	DC1	!	1	A	Q	a	q	1
	STX	DC2	"	2	B	R	b	r	2
	ETX	DC3	#	3	C	S	c	s	3
	EOT	DC4	$	4	D	T	d	t	4
	ENQ	NAK	%	5	E	U	e	u	5
	ACK	SYN	&	6	F	V	f	v	6
	BEL	ETB	'	7	G	W	g	w	7

HIGH ORDER OCTAL DIGITS →	01	03	05	07	11	13	15	17	← LOW ORDER OCTAL DIGIT
	BS	CAN	(	8	H	X	h	x	0
	HT	EM	)	9	I	Y	i	y	1
	LF	SUB	*	:	J	Z	j	z	2
	VT	ESC	+	;	K	[	k	{	3
	FF	FS	,	<	L	\	l	¦	4
	CR	GS	-	=	M	]	m	}	5
	SO	RS	.	>	N	^	n	~	6
	SI	US	/	?	O	_	o	DEL	7

Figure 2.



The first positions of sticks 4 and 6 are respectively "commercial at" and "accent grave." Then the upper & lower case Roman alphabets follow. This offset of one position is historical (from the United Kingdom), and of no importance as long as you remember that it is so.

Following the alphabet in both sticks 5 and 7 are three positions each that one must be very cautious about. In ASCII they are assigned as [, /[1], and] in stick 5 — {, |[2], and } in stick 7. But in the ISO Code and CCITT versions they are reserved for national usage. Table II gives the national use assignment for these positions. Surely you remember that the Scandinavian alphabet has 29 letters, not 26? My friend Orjar Heen in Oslo is very protective of these positions. He says "If you Americans want to sell computers and software abroad, don't use the ASCII characters for these positions in your software."

To be more precise, positions 5/11, 5/12, 5/13, 7/11, 7/12, and 7/13 (noted above) are called *primary* national usage positions. So is 4/0, where ASCII has the "commercial at." Honeywell, for example, uses the "at" in its timesharing systems for deleting the previous character upon entry. But this isn't too serious, because many nations also have the "at" in their primary sets.

	currency		1st 7 national				dia	dia	1st 7 national			dia
	2/3	2/4	4/0	5/11	5/12	5/13	5/14	6/0	7/11	7/12	7/13	7/14
Netherlands–A												
Australia	#											
Belgium–A												
W. Germany–A		$										
US												
Japan					¥							
UK	£											
Italy–A	#	¤										
Switzerland–A		¤										
France–A		$										
USSR		¤										
Netherlands–B												
Belgium–B												
France–B	£	$										
Switzerland–B												
Italy–B	#	$										
Switzerland–C												
Hungary	#	Ft										
W. Germany–B	£	$										
Switzerland–D												
Sweden	#	¤										
Finland												
Denmark												
Norway												
Spain												

Table 2.

Also in sticks 4-7 are three diacritical marks. They are accent grave (`) in 6/0, circumflex (^) in 5/14, and tilde (~) in 7/14. These are called *secondary* national usage positions. In some countries the tilde is a straight overline.

But it is the circumflex where we have a lot of confusion. Teletype first made it an "up arrow" in an earlier version of ASCII, to serve as an exponentiation symbol, primarily for BASIC. But that doesn't do very well, because the exponentiation for FORTRAN is a double asterisk! The FORTRAN version is preferable in France, certainly, because they use such words as crane, cote, cout, and so on.

A companion problem exists in position 5/15, with the underscore. The underscore is neither national nor diacritical; all countries use it just as underscore (and for typesetting it is a U.S. convention to indicate italics, but in Italy it means boldface, except when it is the last character in a line![3]). But Teletype's early version of ASCII used it as a "left arrow" — probably for an assignment symbol equivalent to := in ALGOL. The up and left arrow have been carried over from Teletype into many video terminals. Ask your terminal manufacturer to cease and desist and retrofit. It's not ASCII and will only cause trouble forever.

The last character in sticks 4-7 is the Delete, symbol DEL, in position 7/17. It was put here because the binary code is 1111111, which would be all punched holes in perforated (not always paper!) tape, and that is the only way to make sure that it cannot be misread as some other character. ASCII is a complete set; all positions are assigned to have meaning.

STICKS 2-3

These are usually called the sticks for digits and specials. Remember that they are the "digits" 0 to 9; not numbers, not numerals, not anything but digits! They are in 3/0 through 3/9 so that the low-order 4 bits are the representations for packed decimal. Originally we considered the possibility of a special 4-bit set for numerical applications (see the fifth entry in Table Ia), but it turned out that computer hardware became inexpensive enough to not deprive ourselves of the extra capabilities of the 7-bit and 8-bit sets.

$b_8\ b_7\ b_6\ b_5$ →	0000	0001	0010	0011	0100	0101	0110	0111
$b_4\ b_3\ b_2\ b_1$ ↓ COL / ROW →	0	1	2	3	4	5	6	7
0000	0	NUL	DLE	SP	0	NOTE 1 (α	P	NOTE 1 ′	p
0001	1	SOH	DC1	!	1	A	Q	a	q
0010	2	STX	DC2	″	2	B	R	b	r
0011	3	ETX	DC3	NOTE 1 £ #	3	C	S	c	s
0100	4	EOT	DC4	NOTE 1 $	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
0111	7	BEL	ETB	′	7	G	W	g	w
1000	8	BS	CAN	(	8	H	X	h	x
1001	9	HT	EM	)	9	I	Y	i	y
1010	10	LF	SUB	*	:	J	Z	j	z
1011	11	VT	ESC	+	;	K	NOTE 1 [	k	NOTE 1 {
1100	12	FF	FS	,	<	L	NOTE 1 \	l	NOTE 1 ¦
1101	13	CR	GS	−	=	M	NOTE 1]	m	NOTE 1 }
1110	14	SO	RS	.	>	N	NOTE 1 ^	n	NOTE 1 ~
1111	15	SI	US	/	?	O	—	o	DEL

Note 1

These 12 positions are variable for national usage — 2 for currency, 7 primary national usage, and 3 secondary usage which are diacritical marks when preceded by BSP. The presently-known assignments are given in the table below.

Figure 1.

Position 2/0 is officially called "space." I don't and didn't like it, and would have preferred "blank." Which is why the IBM community often uses a lower case "bee" with a slash through the vertical as its symbol. From the Univac side, the space has the official symbol "delta."

Having mentioned packed decimal, where two digits go into each 8-bit group ("byte" to the American, "octet" to the French), a word of caution on the plus and minus signs — they are in stick 2, rather than stick 3 with the digits. But the low order 4 bits are distinct, and + should be used only as 1011, – only as 1101. I mention this because the nonstandard code EBCDIC permits multiple representations of + and – in packed decimal. And the ASCII representations are not even coincident with any of these, with obvious dangers!

Watch out for the "currency" positions, 2/3 and 2/4. They also have national variations. In ASCII they are customarily # and $, but there are some things to be remembered:

- # is not "number sign" for many countries, most of which use "No." or "Nr." for that purpose. And when it is "number," it must precede the digits, not follow.
- # closely resembled the "sharp sign" in music.
- # is "pound sign" only for the U.S., the only major country still not using the metric system. To the rest, it's kilograms. For now, it's best to use the abbreviation "lb." in the U.S., not the #. In any case, both must *follow* the numeral.
- To the British, a "pound" has the symbol "£", which is why that is the symbol in position 2/3 for the UK. They get very irked when # is called a "pound" sign, especially in software manuals.
- The "dollar" is peculiar to the U.S., Canada, and some others. There are also francs, marks, escudos, pesos, lire, etc.. etc. Which is why the ISO code uses the universal currency symbol in position 2/4. It's a circle with outside spikes at 45, 135, 225, and 315 degrees (O), called "scarab." Table II also shows these assignments for several countries.
- ECMA has provided a separate guideline for specifying international currencies. See the "Where to Get More Information" at the end of this article.

It's a tough problem, and will get worse when we get into expanded character sets for photocomposition and such. For now, all we can do is follow the ASCII standard, which says that # is a "number sign."

Only a few more peculiarities remain for sticks 2-3. An important one is in the double quote, position 2/2, and the single quote, position 2/7. That is, you may think it is a single quote, and even use it so, but it is really an "accent acute" for vowels. It slants from top right to bottom left, to complement "accent grave" in 6/0, which slants from top left to bottom right. Some terminal makers do not realize this pairing, and will have accent grave slanting correctly, but put accent acute as a single quote in the unstylized up and down method. My Terminet is one of those that is OK.

Don't forget that to the typesetter, in contrast to typewriters, both single and double quotes have two forms — opening and closing. In fact, the typesetter gets his double quotes by using two single quotes, of either form, because the quote uses very little space in variable space typesetting. Most terminals, either video or hardcopy, use constant spacing. So double and single quotes must be distinct for that reason.

The last variation is in position 2/6, the ampersand. There are many legitimate different symbols for the ampersand. Neither ASCII nor the ISO Code prescribe any particular one. But this leads us to the next topic — how to represent the ASCII characters in handprinted form, so that they may be input to computer systems.

HANDPRINTING FOR STICKS 2-7

The classical confusion for many years was between the digit zero and the letter "oh," but there are other possibilities for confusion. American Standard X3.45 specifies the handwritten character shapes shown in Figure 3.

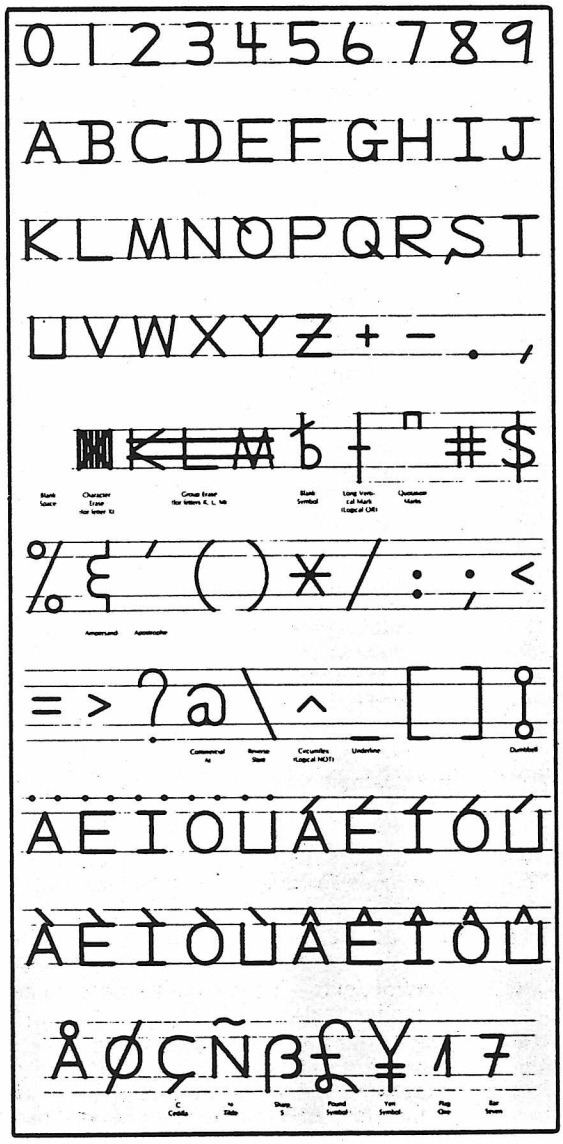


Figure 3.

This clears up a longstanding problem. The communications types, and the armed services, used to put a slash through the zero; somehow the IBM users got to putting the slash through the letter "oh" instead, confusing the Scandinavians greatly. Now it's neither (which helps), just a 180-degree rotation of the letter Q. The earlier German Standard DIN 66 002 prescribed the cursive loop in the upper right, as some may have learned in penmanship courses. It now permits the ANSI form as well.

UPPER AND LOWER CASE LETTERS

Many people are accustomed to using upper case only. This is a hangover from early line printers and limited sets (until the Stretch computer of IBM, characters were usually 6 bits in size). It would have been far better if they had all been lower case in those smaller sets. Putting it simply, would you buy a book to read if it were all

in upper case? Because lower case is much easier and faster to read, lower case should be the default case when one has only the one case. There is no reason why FORTRAN or BASIC processors cannot understand lower case variable names and verbs just as easily as they can understand upper case.

I always recommend getting a terminal with both cases if it is at all affordable. Second best is making sure that a single-case terminal is retrofittable later, if necessary. And if a single-case terminal, get it in lower case only, if possible. There has been much reportage in the computer trade press about eyestrain resulting from using computer terminals. Is the reason obvious?

STICKS 0, 1

These are the control characters. The most important distinction in ASCII is the split between sticks 0-1, Controls, and sticks 2-7, Graphics. We'll see this later on in the standards for Code Expansion (to 8 bits or more), and Code Extension (alternate sets, such as Cyrillic for the USSR, and Kata Kana for Japan).

Unfortunately, there is, despite the standard, much difference between the ways that various terminal devices handle these control characters. They may act differently, or they may not be operative at all. I have two very useful programs, written in the TEX language (Reference 6). One lists each symbol by name and then shows its action between parentheses. The other asks you to depress in turn all the funny keys on your terminal, and then tells you what control character(s) they generate. if any.

GRAPHICS FOR THE CONTROLS

There are standard graphical representations for the 32 controls, space, and delete. They are defined by ISO 2047, American Standard X3.32, and ECMA-17, and are shown integral to Figure 1. Some terminals are advertised as ASCII terminals, and yet generate Greek or other characters for these positions. Don't believe it! These symbols are every bit as useful as any Greek characters could be.

There are five groups in the basic control set.

STICKS 0, 1 — Logical Communication Control (10)

This group is used for both communication and for labeling of media. It includes:

SOH (0/1) (Start of Heading) — used as the first character in the heading of an information message.

STX (0/2) (Start of Text) — terminates the heading just before the text.

ETX (0/3) (End of Text) — Last character in the text message. Unfortunately, it is generated on many terminals via Control-C, and that's just to the right of Control-X on the keyboard, which is commonly used to cancel a bad input line. And if you mis-key — ouch!

EOT (0/4) (End of Transmission) — the last character in any transmission, and usually it turns your device off!

ENQ (0/5) (Enquiry) — requests a response from a remote station, either an identification of that stations (Who are you?) or its status.

ACK (0/6) (Acknowledge) — used by a receiver to reply "yes" to a sender.

DLE (1/0) (Data Link Escape) — an Escape character, especially for communications, analogous to ESC (1/11). It signals the start of a character sequence that causes a shifting into another set of communication controls, whenever they are needed.

NAK (1/5) (Negative Acknowledge) — used by a receiver to reply "no" to a sender.

SYN (1/6) (Synchronous Idle) — needed by synchronous transmission systems to get into, or stay in, synchronization when no other such signal is available to them.

ETB (1/7) (End of Transmission Block) — indicates the end of some division of data that the transmission system must make, unrelated to any division in the format of the logical data itself.

One lists each symbol by name and then shows its action between parentheses. The other asks you to depress in turn all the funny keys on your terminal . . .

STICKS 0, 1 — Physical Communication (4)

This group is used for communications. It includes:

NUL (0/1) (Null) — the standard says that it is "used" to accomplish media fill or time fill" . . . "may be inserted into or removed from a stream of data without affecting the information content of that stream." And that's exactly what the standard also says about DELete (7/15), which it lists as a control character even though it is not in the control sticks! The only difference I can see between them is that on perforated tape you can make any character into a DELete, but none into a Null.

CAN (1/8) (Cancel) — the receiver is to disregard the data received up to that point, starting from restart point that receiver and sender have agreed upon. It is common in timesharing for Cancel (often generated by a Control-X) to work on a line-at-a-time basis, to delete an unwanted string of entry characters, and effectively put one back to the position of reentering the entire line. In this case, the agreement between sender and receiver is "back to the last CR." But there are many other ways that Cancel could be used, and for parallel as well as serial transmission.

SUB (1/10) (Substitute) — a character that says probably we would have had another character in this position if we could have figured out what it was supposed to be! There are many reasons for such confusion — perhaps parity didn't check out. But it is better to put in a SUB to keep the field lengths and such correct. Moreover, note its symbol, a mirror image (not the Spanish inverted) question mark. If this is displayable, it will tell you definitively that the system doesn't know what it is, and you can make a good guess in many cases, particularly in word text.

EM (1/9) (End of Medium) — defines the previous character as the last usable character on that medium, whether or not there is more recordable space on the medium.

STICKS 0, 1 — Device Control (11)

This group is used for control of devices such as terminals.

HT (0/9) (Horizontal Tabulation) — the standard says that is "advances" the active position to the next predetermined character position on the same line." There are two ways this can work:

1. Right at the terminal, if it has the horizontal tab capability built in. Sometimes you can set the tab positions by using the terminal only; almost always the computer can be made to set the tabs on the terminal. Then when you hit HT during entry, or HT is read from the computer output, the printing or displaying (active) position will skip to the next tab setting.
2. By a formatting program in the computer, which must be given some indication of the tab setting positions in force at any particular point in the file. The program then simulates horizontal tab movement by filling the lines with spaces as needed to achieve the alignment.

VT (0/11) (Vertical Tabulation) — the standard says that it "advances the active position to the same character position on the next predetermined line." And if you agree with somebody else, it can be to the first position in that line instead. This is a very dangerous character to use. It cannot be used directly on any terminal that I know of. Even if it could, the implementation rules are not supplied unambiguously in the ASCII standard. And for use by a formatting program, one would have to predefine the number of lines to be skipped. That's pretty tough when you are inserting and deleting lines, as every programmer knows.

LF (0/10) (Line Feed) — like vertical tab, but just to the next line, which is clean enough. If receiver and sender agree (again as in vertical tab), it can be to the first position of the next line, in which case it is called New Line (NL). Some manufacturers implement this. I personally prefer having a separate Carriage Return and Line Feed. Both codes can be generated with a single keystroke, and they often are.

FF (0/12) (Form Feed) — again like vertical tab, to the same cahracter position unless sender and receiver agree that it is to the first position in the new line, except that the tab is to a new line position that is related to a form of some size (those that fold 11 inches apart, for example). This control could run wild if your terminal or other display device is not equipped to handle it, so use it with caution in files.

CR (0/13) (Carriage Return) — moves the active position to the first position on the *same* line! Not like typewriters. They have effectively incorporated the New Line feature. But the non-advancing CR is better for terminals, even if it is misnamed. Neither video terminals nor ball and daisy wheel typewriters have carriages, so live with it.

BS (0/8) (Backspace) — Backspace is a very tricky character. On some terminals, such as video terminals, there is no key to generate Backspace for entry into the text stream or buffer. On many it can be created via Control-H. Even then, it may or may not be operative.

Backspace is meant for physical movement of the active position (which may or may not coincide with a cursor position, when such exists). Historically, it was included for hardcopy terminals and other hardcopy devices for some of these uses:

- Underscoring (underlining).
- Other forms of highlighting, such as bold. For example, the sequence A BS A BS A would strike the A three times on a hardcopy device, and make it look boldface (such a sequence can also be translated to call a boldface font in photocomposition).
- Editing indications. For example, in legislative bill drafting to indicate the deleted or changed portion:
  This is ~~obsolete~~.
- Forming composite characters, e.g.:
  ¢ ± ≠ ↑ ½ ¾ £ (Hungarian forint)
- Forming accented letters, primarily for European languages. Examples:
  Ä Å Ø (Scandinavian letters following Z)
  Ñ à ã ô ü

Warning: Backspace is entirely different from a cursor movement on a video terminal! When the cursor is moved to a position where a character is already entered, succeeding entry in that position usually destroys the original character and replaces it with the new entry.

I personally haven't seen any video terminals with a true backspace. A former president of Infoton told me it could be done as an engineering special for about $5,000 one-time cost.

Warning: There are three ways to create underscored text for hardcopy terminals:

1. The characters, that many backspaces, and that many underscores (or vice versa).
2. A character, BS, underscore, the next character, etc. This is called the canonical form, and is used quite commonly.
3. Underscore, BS, character, underscore, etc.

I have noticed a lot of difficulty moving back and forth between hardcopy (at my home) and video (in my office) terminals. One tends to underscore on the hardcopy terminal and forget that half of the pairs are going to be wiped out by the cursor on the video terminal. In the first two methods above, it's the text that gets wiped out, and it's hard to read on the fly. So if you plan to display a file on a video terminal, find another highlighting method, or use the third underscoring convention. Even that may give problems if done by embedding an underscoring command in the file you pass to a formatting program; most such programs put the underscore last instead of first.

BEL (0/7) (Bell) — sounds an audible signal to get the user's attention. Some terminals are not so equipped, but they should be. It's good human engineering. But please give me an adjustable volume control!

And then there are the four device controls for unspecified purposes, DC1, DC2, DC3, and DC4 — in positions 1/1 through 1/4. Different manufacturers treat these like a wild card in poker — they make them anything that they want. Doesn't lead to much compatibility, so beware.

STICKS 0, 1 — Field Separators (4)

This group is used for formatting and string processing. These are the separators in positions 1/12 to 1/15. I got the idea originally from the Word Mark in the IBM 1401, which used an extra bit in the low-order character in a field as a delimiter. ASCII uses special and separate characters to indicate a hierarchical structure. Originally I put in eight such characters, but only these four remain:

FS(FileSeparator — 1/12)
GS(GroupSeparator — 1/13)
RS (Record Separator — 1/14)
US(UnitSeparator — 1/15)

FS is most inclusive, US the least inclusive. And we can consider the blank/space as the next lower order separator from these. Suppose we had a line of text like this:

(text1)US(text2)US(text3)RS(text4)US(text5)GS(text6)

On many terminals these delimiting control characters would not print, so we would see only a continuous stream. On others they might show as spaces. A TEX command to break the line at the record separator would be:

scan:line:*rs

The variable *left would contain "(text1) . . . (text3)". The variable *right would contain "(text4) . . . (text6)".

STICKS 0,1 — Changing Sets (3)

This group is used for moving to and from alternate graphic and control sets. This includes ESCape (1/11), Shift Out (0/14), and Shift In (0/15).

These basic control characters have permitted design of a quite marvelous structure for extension and expansion. It allows us to code and classify most of the world's graphic symbols for computer storage, interchange, and display. This big area will form most of Part III of this article.

IN THE NEXT INSTALLMENT

The ASCII Collating Sequence

ASCII and Programming Languages

ASCII and Media

Keyboards

ASCII and Display/Printing

Code Extension — Alternate Controls

Code Extension — Alternate Graphics

ASCII and Non-Latin Alphabets

Code Expansion — 8-bit ASCII

WHERE TO GET MORE INFORMATION

There are four sets of Information Processing Standards that may be of concern to you:

- ISO. Sold only through ANSI (American National Standards Institute), which has the franchise. That makes the prices high — much higher than in other countries.
- ANSI. These are American National Standards developed via the X3 and X4 committees, mostly. Prices still pretty high.
- ECMA (European Computer Manufacturers Association), 114 Rue du Rhone, 1204 Geneva, Switzerland). Free, and they have a lot more advanced standards than ISO and ANSI. But a modest donation would not be unwelcome.
- Your friendly U.S. Government, in the person of the Department of Commerce, National Bureau of Standards, Institute for Computer Sciences and Technology, in Gaithersburg, MD 20760. If by any chance you are employed by the U.S. Government, you get FIPS PUBS (Federal Information Processing Standards Publications) for cheap. Otherwise, see ANSI. (Refer to Tables 1a, 1b, and 1c). In many cases they are essentially reprints of the ANSI standards, for a fraction of the cost.

If you can't wait for the standards to be approved and published, catch them in progress. Ask CBEMA, the sponsor of ANSI X3, to put you on an observer list for the committee in your area of interest. The address is:

Robert Brown, Director of Standards
Computer & Business Equipment Manufacturers Association
1828 L Street NW
Washington, D.C. 20036
(202) 466-2288 Telex 89 29042

REFERENCES

1. ANS X3.4-1977, available from the American National Standards Institute, 1430 Broadway, New York, NY 10018.
2. ISO 646, avialable from ANSI (Reference 1).
3. R.W. Bemer, "ASCII — the data alphabet that will endure," in *Management of data elements in information processing,* National Bureau of Standards, 1975 October, 17-22.
4. R.W. Bemer, "A view of the history of the ISO character code," Honeywell Computer J. 6, No. 4, 1972, 274-282.
5. E.H. Clamons, "Character codes: who needs them?", Honeywell Computer J. 5, No. 3. 1971, 143-146.
6. The TEX Subsystem of the Timesharing System, Series 60 Level 66, Honeywell Information Systems, 200 Smith Street, Waltham, MA 02154, Order DF72.

ACKNOWLEDGEMENTS

Thanks go to co-workers at Honeywell Information Systems: Eric Clamons for much background, insight, and experience gained from working for a long time as chairman of X3J2 — the committee charged with the development of ASCII. And to Pat Skelly, ACM representative on ANSI X3, for collecting all the various national and international standards documentation upon which many of the figures were based.

FOOTNOTES

[1]For those curious about the reverse slash, it came from ALGOL 58. The reference language specified ∧ and ∨ as the symbols for AND and OR respectively. I put the reverse slash in so these could be made as 2-character groups — and

[2]You will still see many terminals where this vertical bar is broken in the middle. This resulted from a hassle with the PL/I people, who wanted to stylize the exclamation point (2/1) as a vertical bar for OR in that language. And of course that would make the graphics the same. The compromise (at horrendous cost in people time) was to break the real vertical bar in ASCII. But it turned out that the PL/I people didn't really need it, or else it gained no momentum, so the real vertical bar is back to normal in ASCII-1977. Let's fix those terminals.

[3]The Italians also have a different solution to hyphenation and right justification. It ignores the syllable structure and simply demands that if, when you get to the last position in the line, the current word is not yet completed, that last character shall be underscored, and the word continued without fuss on the next line. I rather like it.

THE FATHER OF ASCII, Robert W. Bemer

Robert "Bob" Bemer received his A.B. in Mathematics from Albion College in 1940, and a Certificate in Aeronautical Engineering from Curtiss-Wright Tech a year later.

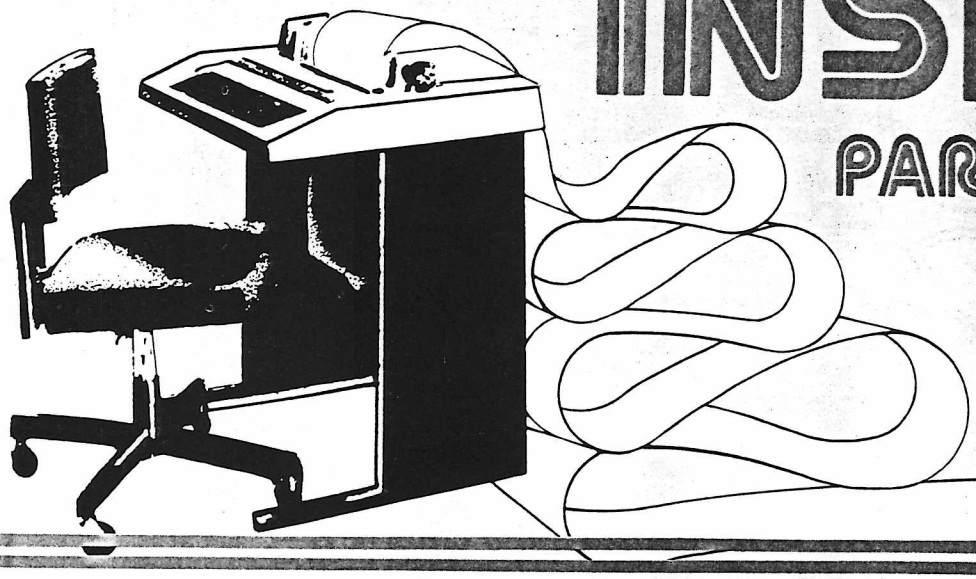
His vast work experience includes employment with the major leaders of the aircraft and electronics industries—most recently, as Senior Consulting Engineer with Honeywell Information Systems.

Highlights of his many accomplishments include: The discovery of polynomial telescoping (1954); creation of the PRINT 1 programming system for IBM (1956); development of FORTRANSIT; development of COMTRAN, one of the three major inputs to COBOL; development of XTRAN, predecessor to ALGOL (1958); was a major influence in the choice of the 8-bit character in IBM System 360 (1960); an influence in building the 1108 and 6000 systems; and editor of Honeywell Computer Journal.

He has an impressive list of over 71 publications to his credit.□

INSIDE

PART 2 OF



The abstract aspects of ASCII were treated in Part I. Now we come to some aspects of usage and implementation. Certainly one major use area is the ordering of files.

THE ASCII COLLATING SEQUENCE

To put items in some ordering, the entire precedence relationship for that ordering must be defined. Higher or lower, precedes or follows, or whatever. For single characters, this ordering relationship is called the "collating sequence".

The ASCII standard used to say that the collating sequence for both graphics and control characters is defined simply by their binary representations. Later it added a warning that this collating sequence "cannot be used in many specific applications that define their own sequence". What an understatement!

The 1977 version hedges and speaks all around the problem without making it clear. It's not all that difficult. Suppose you have two files, and you want to know how they differ and/or how they are the same. For this purpose, the implied collating sequence (straight binary comparison) is just fine. The two files will be in the same order, and can be matched.

Whether that straight binary ordering can be used for any other purpose is doubtful. It won't work for signed numbers.

Ordering Numerals

Take these four values: 22, 13, minus 6, and minus 31. If the sign is placed before the digits, ordering by the ASCII collating sequence yields:

+13
+22
−06
−31

This is obviously worthless. It's because ordering is decided left to right, and the minus sign has a binary value 2 higher than the plus sign. Or if the sign were to follow the numeric values we would get:

06−
13+
22+
31−

because the complete decision is made in the leading digit. Again, a worthless sequence.

The way to achieve a proper ascending sequence is to separate the values into two groups, ordering those with plus signs in ascending sequence, and those with minus signs in descending sequence. Then put the plus group following the minus group. And vice versa for a total descending sequence. Notice that this works regardless of whether the sign precedes or follows the digits.

Ordering Alphabetic Fields

Alphabetic ordering is even more complex, particularly in handling both upper and lower case. Again the implied ASCII collating sequence can go wrong. People who have not studied the collating problem for data containing both upper and lower case are inclined to jump to wrong conclusions. I did myself, for the IBM Stretch computer in 1958, assigning the ascending binary sequence as AaBbCc. Using this for a telephone directory would give us the lefthand column. The straight binary sequence of ASCII would yield the righthand column, just slightly different:

De Carlo	De Carlo
De La Rue	De La Rue
De Long	De Long
DeLair	DeLaRue
DeLancey	DeLair
DeLaRue	DeLancey
Delancey	Delancey
de Carlo	de Carlo
de la Rue	de la Rue
deLancey	deLancey

Either version will get a lot of anguished subscribers!

In the simplest case, two alphabetic items must be compared with the case ignored. Only if they are *then* equal is case called into consideration to break the tie, and it is also applied successively left-to-right!

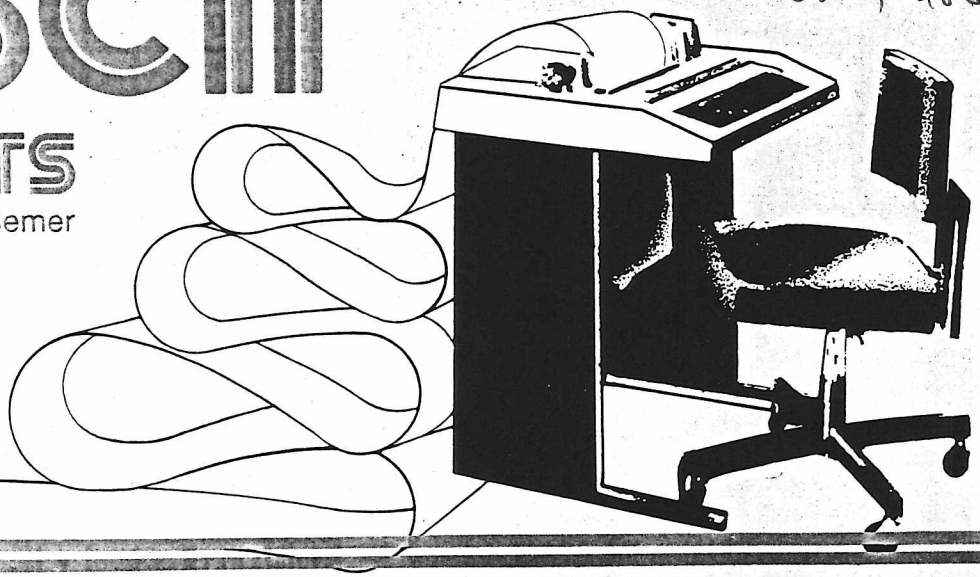
In short, the upper and lower case versions of a letter do not both get full graphic significance. Typing either "Y" or "y" will indicate a "yes" reply, but "N" will not. Because the case distinction is minor, comparisons must first be made on major distinctions, with the minor distinctions used only as tie-breakers. Accenting of let-

ASCII

3 PARTS

By R. W. Bemer

Interface Age June 1978 vol 3 no 6



ters must also be considered minor, if accomplished via backspace, but this leads us into rules controlled by foreign governments, and won't be considered here.

Real life is more complicated than this. The ordering and sequencing of characters and words cannot always be accomplished by simple binary comparison of codes. There are constructions such as O'Reilly, l'Informatique (as data processing is called in French), and Smith-Jones — to say nothing of the Juniors, IIIs, Esq., FBCS (which I am), and so on.

Making an ASCII comparison, with the case as a minor, gives us:

De Carlo
de Carlo
De La Rue
de la Rue
De Long
DeLair
DeLancey
Delancey
deLancey
DeLaRue

Because we at first ignored case here, De Carlo and de Carlo have identical bit patterns. Tie-breaking is done by appending the binary pattern representing case, "0" for upper, "1" for lower. Specifically, 01001111 for De Carlo, 11001111 for de Carlo.

	D	E		C	A	R	L	O	
De Carlo	44	45	20	43	41	52	4C	4F	(4F)
de Carlo	44	45	20	43	41	52	4C	4F	(CF)

But even this method will not put "DeLaRue" and "De La Rue" in the same cluster. And surely this is desirable and even mandatory. It will require some special handling for spaces. The New York Telephone Company's document on this problem runs to several pages! They'd probably give you a copy upon request. You might need to know those rules before trying one of the toughest acts in data processing — putting last name first, or vice versa.

Using Controls in Ordering

There is one more aspect of ASCII useful to the order-

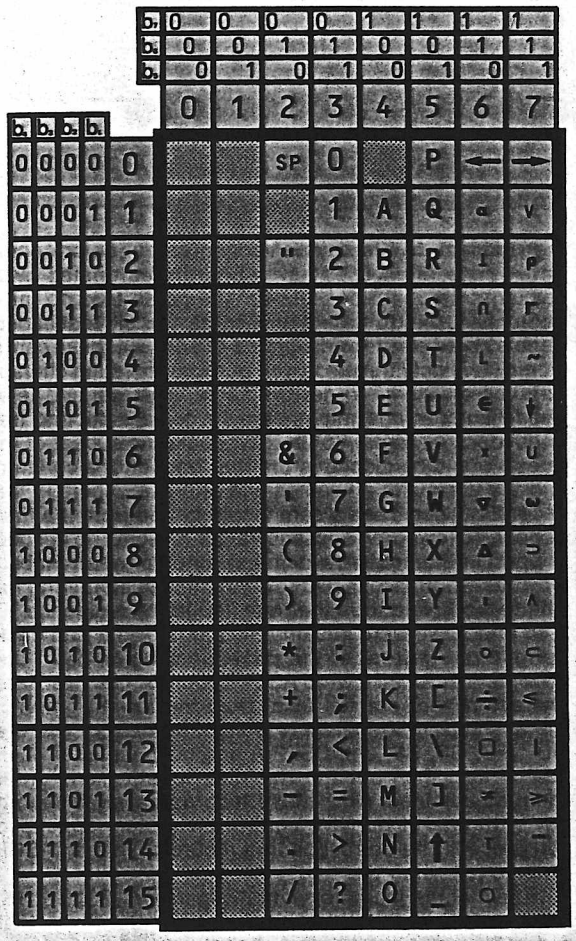
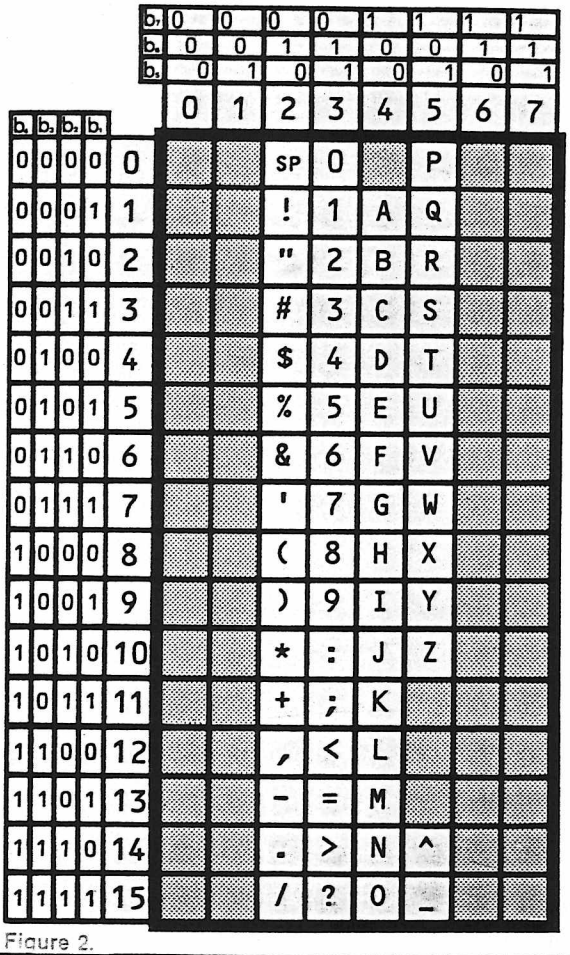
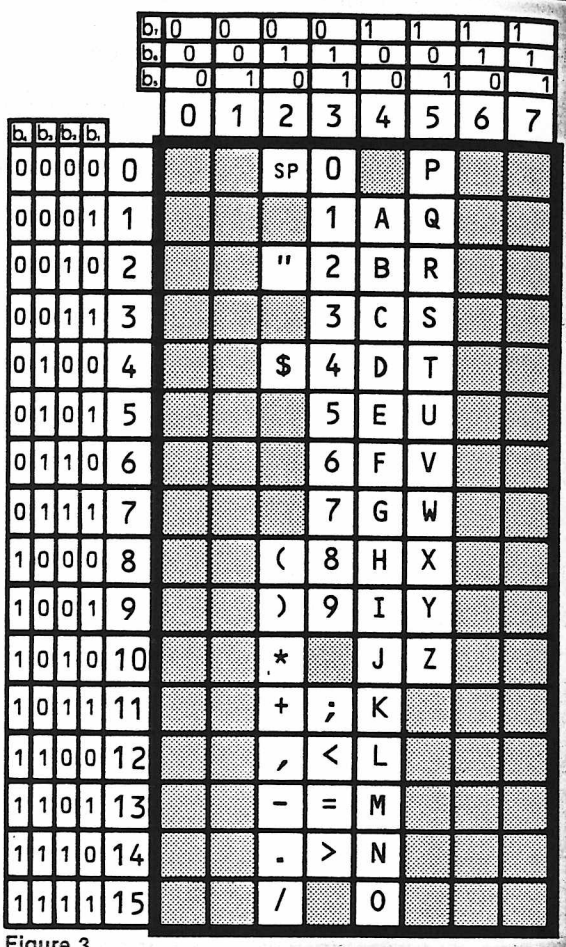


Figure 1.



b_4 b_3 b_2 b_1	b_7 0	0	0	0	1	1	1	1
	b_6 0	0	1	1	0	0	1	1
	b_5 0	1	0	1	0	1	0	1
		0	1	2	3	4	5	6	7
0 0 0 0	0			SP	0		P		
0 0 0 1	1			!	1	A	Q		
0 0 1 0	2			"	2	B	R		
0 0 1 1	3			#	3	C	S		
0 1 0 0	4			$	4	D	T		
0 1 0 1	5			%	5	E	U		
0 1 1 0	6			&	6	F	V		
0 1 1 1	7			'	7	G	W		
1 0 0 0	8			(	8	H	X		
1 0 0 1	9			)	9	I	Y		
1 0 1 0	10			*	:	J	Z		
1 0 1 1	11			+	;	K			
1 1 0 0	12			,	<	L			
1 1 0 1	13			-	=	M			
1 1 1 0	14			.	>	N	^		
1 1 1 1	15			/	?	O	_		

Figure 2.



b_4 b_3 b_2 b_1	b_7 0	0	0	0	1	1	1	1
	b_6 0	0	1	1	0	0	1	1
	b_5 0	1	0	1	0	1	0	1
		0	1	2	3	4	5	6	7
0 0 0 0	0			SP	0		P		
0 0 0 1	1				1	A	Q		
0 0 1 0	2			"	2	B	R		
0 0 1 1	3				3	C	S		
0 1 0 0	4			$	4	D	T		
0 1 0 1	5				5	E	U		
0 1 1 0	6				6	F	V		
0 1 1 1	7				7	G	W		
1 0 0 0	8			(	8	H	X		
1 0 0 1	9			)	9	I	Y		
1 0 1 0	10			*		J	Z		
1 0 1 1	11			+	;	K			
1 1 0 0	12			,	<	L			
1 1 0 1	13			-	=	M			
1 1 1 0	14			.	>	N			
1 1 1 1	15			/		O			

Figure 3.

ing problem. In the days of punch cards, before computers, one often used several card files related by a key. A sorter (with pockets for the cards to drop into) might be used to select the cards for all redheaded females between 18 and 24 years of age. But these cards would have only the employee number and such characteristics on them. To get the name, address, and telephone number one might have to go to a second (related) deck of cards. So the first deck (the subset of interest) would be placed in the first hopper of a collator, and the deck with all names and phone numbers in the second hopper. Then a card would be fed from the first hopper, followed by successive cards from the second hopper, until a match was found on employee number. Obviously both decks had to be in the same ordering for this to work, and thus the term "collating sequence".

In effect, we were sticking the cards of the first deck upright just in front of the corresponding cards of the second. To do this with ASCII requires that we have characters that collate lower than the lowest graphic, the space (2/1). We do have them. The best to use are NUL, FS, GS, RS, and US. Put one of these after each search key, then put the two files together and order them as adjoined. Now those records having a search key with one of our five control characters appended will precede the corresponding record having an ASCII graphic following the key.

Note that the four information separators (FS, GS, RS, US) are designed to collate just behind Space, in that order. This contiguity means that they can be used as a hierarchy of spaces of different class.

Other Collating Features

ASCII was designed when there was substantial investment in files already ordered on a Topsy-class IBM sequence, where the basic punctuation was low to the alphabet, but the digits were high to it. How then to accommodate this and still provide a 4-bit subset? My morning shower provided a solution (it still does!).

The 4-bit subset is formed of the first 10 graphics of stick 3 (the digit graphics) and the last 6 of stick 2. This jog was shown shaded in the early forms of ASCII, but has all but disappeared from memory now. It enables stick 3 (with the digits and new special graphics) to be ordered high to all the others via passive logic, thus overcoming opposition to the adoption of ASCII.

ASCII AND PROGRAMMING LANGUAGES

Standard ECMA-53 (1978 Jan), "Representation of Source Programs for Program Interchange," gives the subsets and/or modifications of ASCII as they are used for these five programming languages (Footnote 1):

	NO. OF CHARACTERS USABLE	
Language	Subset of ASCII	Other
APL	57	32
Minimal BASIC	60	0
COBOL	51	0
FORTRAN	49	0
PL/I	55	2

Figures 1 through 5 are the character sets for these languages as given in ECMA-53. They show the only characters permissible for use in source programs, except for:

non-numeric literals	in COBOL
comment-entries	"
comment lines	"
character constants	in FORTRAN
comments	"

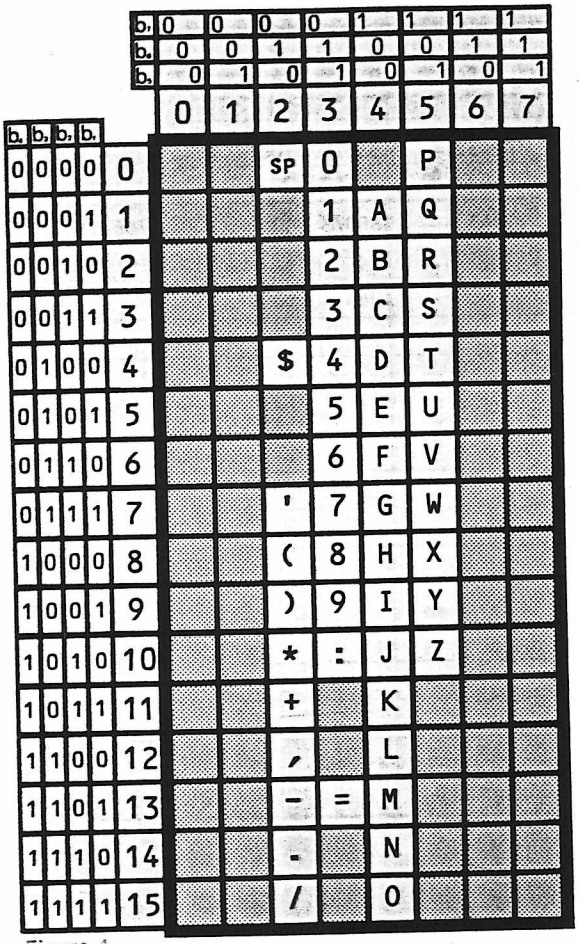


Figure 4.

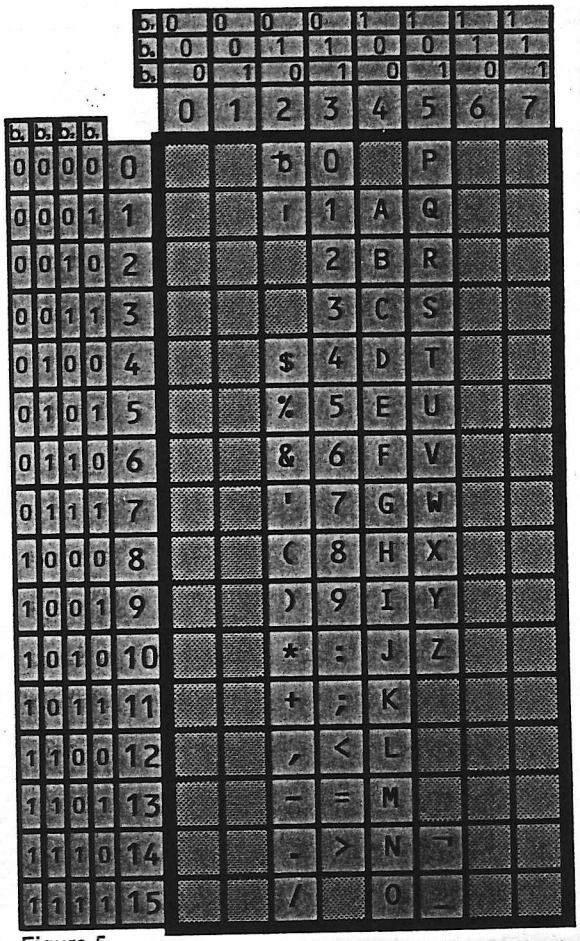


Figure 5.

character-string-constants in PL/I
comments "

For these purposes only, other ASCII characters may be used, providing there is agreement between the sender and receiver for any interchange of source programs.

The TEX language has gone farther than this general caution. There the specific characters have permanent names. For example, one could say:

linefeed = "
" (actual line feed inside the quotes)

if If:eqs:linefeed ....

and it would be true, because "*lf" is the permanent name of Line Feed. The control characters have names that are the letters from the ASCII chart, preceded by the asterisk to show that they are read-only variables with permanent content. TEX can in fact operate upon all 256 characters of ASCII in an 8-bit byte, all 512 in a 9-bit byte.

Specific Notes on the Figures

APL -Sticks 6 and 7 (ordinarily lower case alphabet) are replaced entirely except for the DELete position.
-Space is nonprinting, although the symbol shown is SP.
-Ampersand (2/6) is not used for writing source programs, except as the last character of a line if that line is to be continued on the next line.

PL/I -In position 2/1, the exclamation point is replaced by a vertical bar for OR.
-In position 5/14, the circumflex is replaced by the symbol shown, for NOT.
-If you have to use your terminal for both PL/I and some other programming language, forget that foolishness. You can get by with the exclamation point as OR, and the circumflex as NOT. The important point in source program interchange is to have the encoded representations of the characters exchanged correctly.

(all) -Although the character BLANK (space) is shown as the flagged lower case "b" in the FORTRAN and PL/I sets, there is no printing graphic to indicate it. For all practical purposes, it is really the Space of ASCII (2/0).
-Four of these five languages (not APL) have the "$" shown in 2/4. When the International Reference Version of the code is used, this becomes the universal currency symbol, which is also acceptable.
-Minimal BASIC uses "#", which is the International Reference Version symbol. The national symbols, such as the English pound sign, are also acceptable.

ASCII AND MEDIA

ASCII and Punch Cards

Reading and punching equipment for punch cards, being very mechanical, is so expensive that microcomputer people are unlikely to use them. So you might ask why we bother here with the representation of ASCII on this medium? I can think of at least three reasons:

- A scientist at the U.S. National Bureau of Standards said once that if punch cards were on the way out, it was the only product he ever saw dying on an upward usage curve. Thus they are likely to be around for a long

	ISO	ECMA	ANSI	FIPS PUB	CSA	BS	AS	CCITT	JIS	GOST
Hollerith Punched Card Code	1679 2021	44	X3.26-1970 $4.25	14	Z243.14 .36	4636/3 /4	1063			
Track Assignment - 25.4 mm Perf. Tape	1113	10	X3.6-1965 $3.00	2	Z243.8	3880/3	1062		C6221	
Track Assignment - 12.7 mm Mag Tape 200 cpi NRZI 9-track	1862	5	X3.14-1973 $3.25			3968	1008			
Track Assignment - 12.7 mm Mag Tape 800 cpi NRZI 9-track	962 1863	12	X3.22-1973 $3.75	3-1		4503/1	1009		C6222	
Track Assignment - 12.7 mm Mag Tape 1600 cpi PE 9-track	3788	36	X3.39-1973 $3.75	25		4503/2				
Track Assignment - 12.7 mm Mag Tape 6250 cpi GCR 9-track	DP 5652		X3.54-1976 $5.25	50						
Labeling & File Structure - 12.7 mm MT	1001	13	X3.27-1977 (unpriced)		Z243.7	4732	1068			
Track Assignment - Magtape Cassette 3.81 mm, 32 bpmm	3275 3407	34	X3.48-1977 $5.75	51		5079/1				
Labeling & File Struct. - 3.81 Magtape Cassette	DIS 4341	41								
Track Assignment - 6.35 mm Cartridge Tape 64 bpmm PE	DIS 4057	46	X3.56-1977 $4.24							

Table 1a. Standards for ASCII on Physical Media.

	ISO	ECMA	ANSI	FIPS PUB	CSA	BS	AS	CCITT	JIS	GOST
Bit Sequencing in Serial Transmission			X3.15-1976 $3.00	16-1				V.4 X.4		
Char. Structure & Parity Sense - Serial-by-Bit			X3.16-1976 $3.50	17-1				V.4 X.4		
Char. Structure & Parity Sense - Parallel-by-Bit			X3.25-1976 $3.50	18-1				V.4 X.4		
Procedures for Using Commun. Control Chars.	1745	16	X3.28-1976 $10.50		Z243.13	4505/1	1484/1			
Message Heading Formats	1745		X3.57-1977 $5.25							
Advanced Data Commun. Control Procedures			BSR X3.66							

Table 1b. Standards for ASCII in Communications.

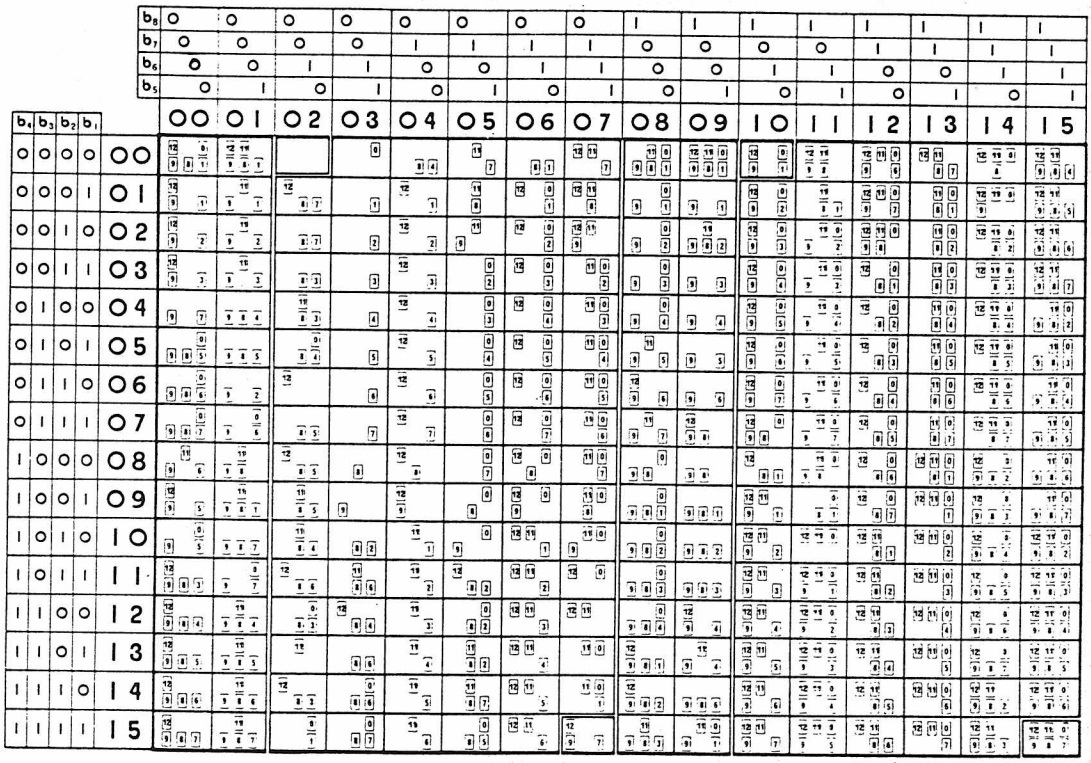


Figure 6.

	12	11	0	12 0	12 11	11 0	12 11 0
	02/06	02/13	03/00	02/00	07/11	07/12	07/13	11/10
1	04/01	04/10	02/15	03/01	06/01	06/10	07/14	13/09
2	04/02	04/11	05/03	03/02	06/02	06/11	07/03	13/10
3	04/03	04/12	05/04	03/03	06/03	06/12	07/04	13/11
4	04/04	04/13	05/05	03/04	06/04	06/13	07/05	13/12
5	04/05	04/14	05/06	03/05	06/05	06/14	07/06	13/13
6	04/06	04/15	05/07	03/06	06/06	06/15	07/07	13/14
7	04/07	05/00	05/08	03/07	06/07	07/00	07/08	13/15
8	04/08	05/01	05/09	03/08	06/08	07/01	07/09	14/00
9	04/09	05/02	05/10	03/09	06/09	07/02	07/10	14/01
8 2	05/11	05/13	05/12	03/10	12/04	12/11	13/02	14/02
8 3	02/14	02/04	02/12	02/03	12/05	12/12	13/03	14/03
8 4	03/12	02/10	02/05	04/00	12/06	12/13	13/04	14/04
8 5	02/08	02/09	05/15	02/07	12/07	12/14	13/05	14/05
8 6	02/11	03/11	03/14	03/13	12/08	12/15	13/06	14/06
8 7	02/01	05/14	03/15	02/02	12/09	13/00	13/07	14/07

12	11	0		12 1	12 11	11 0	12 11 0	
10/08	11/01	11/09	06/00	12/03	12/10	13/01	13/08	8 1
00/01	01/01	08/01	09/01	10/00	10/09	09/15	11/11	9 1
00/02	01/02	08/02	01/06	10/01	10/10	11/02	11/12	9 2
00/03	01/03	08/03	09/03	10/02	10/11	11/03	11/13	9 3
09/12	09/13	08/04	09/04	10/03	10/12	11/04	11/14	9 4
00/09	08/05	00/10	09/05	10/04	10/13	11/05	11/15	9 5
08/06	00/08	01/07	09/06	10/05	10/14	11/06	12/00	12 6
07/15	08/07	01/11	00/04	10/06	10/15	11/07	12/01	12 7
09/07	01/08	08/08	09/08	10/07	11/00	11/08	12/02	9 8
08/13	01/09	08/09	09/09	00/00	01/00	08/00	09/00	9 8 1
08/14	09/02	08/10	09/10	14/08	14/14	15/04	15/10	9 8 2
00/11	08/15	08/11	09/11	14/09	14/15	15/05	15/11	9 8 3
00/12	01/12	08/12	01/04	14/10	15/00	15/06	15/12	9 8 4
00/13	01/13	00/05	01/05	14/11	15/01	15/07	15/13	9 8 5
00/14	01/14	00/06	09/14	14/12	15/02	15/08	15/14	9 8 6
00/15	01/15	00/07	01/10	14/13	15/03	15/09	15/15	9 8 7

Figure 7.

time, and you may need to transfer some of those files to other media that you do use.

- There is some likelihood that microcomputers could be used in the reading and punching equipment itself, to make it less expensive.
- ASCII users are going to be confronted for a while yet with one of the several versions of IBM's EBCDIC, and the punch card assignments provide the only legitimate link for conversion of EBCDIC files to ASCII.

So Figure 6 defines the hole patterns for the binary encodings. And Figure 7 defines the encodings for the hole patterns. Don't worry about the inconsistency in the relationships. Nothing can be done about it now, because it started with Herman Hollerith's first U.S. Census machines in 1890. At first only digits and + and − signs were used. Then the code was expanded to the upper case alphabet. And other special characters for commercial use. When FORTRAN came along in 1964, it turned out that the limited capability of the subset of a 6-bit set would not permit the graphics needed for scientific work. For a long while there were dual graphic representations for several of the punch card code combinations, and this carried over into printer chains, and so on.

The only logic that the patterns follow is that they do or do not have a punch from among these six possibilities:

12-punch (top row)
11-punch (next to the top row)
0-punch
8-punch
9-punch (bottom row)
a punch from among the digits 1 through 7

Including the no-punch-at-all combination (NUL), this gives 256 combinations, just right for the 8-bit code. Although ASCII was technically only a 7-bit code at the time this rule was formulated, it was felt necessary to plan ahead a little.

ASCII and Magnetic Tape

Figure 8 gives a compact representation of several relationships, among which is the assignment of ASCII bit pattern to 9-track magnetic tape. The jumbled assignment may remind you of the "firing order" for the cylinders of an automobile engine. In fact, we used to call it just that. It was intentional for increased reliability. As in so many cases, better technology has removed the need for peculiar design, but the assignments are unchangeable because of data file investment.

There is no parallelism in recording and reading on cassettes and cartridges. The ASCII bits are recorded serially in the track. Thus Figure 8 does not consider these media.

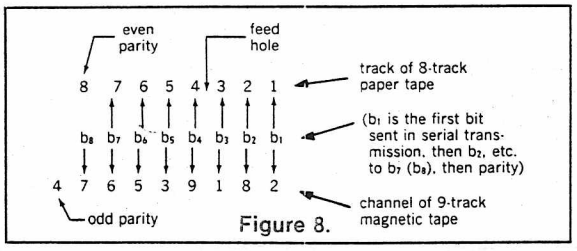


Figure 8.

ASCII and Communications

Not only is the topic of ASCII and communications a very complex and large dissertation for this article — it is also undergoing substantial rethinking, enlargement, and invention. You will have to follow on your own the workings of the CCITT, the various networking systems of the several large and many small manufacturers of computer systems, and the offerings of the common carriers — either on the local distribution system (via ATT) or direct distribution (via Satellite Business Systems).

Many of the existing standards are listed in Table 1b. Many more are under development. Arguments are raging internationally on the merits of packet switching, byte protocols, value-added systems, open-working systems, tariffs, data movement across national borders, the X.25 protocol, etc., etc. ATT is offering a new service because they suddenly discovered data-undervoice (DUV). All I can tell you now is that it is all based upon ASCII, and the proposed protocols are all dependent upon the ASCII control characters in stick 0 and 1. It will take years for this to shake out, and for now all one can do is get on the CBEMA mailing list (see reference, Part I, INTERFACE AGE, May, 1978).

ASCII AND THE METRIC SYSTEM

The full ASCII graphic set (both cases) is sufficient to indicate all symbols and prefixes of the SI (International System of Units, the new metric system), with three exceptions. They are the Greek letters "omega" for "ohm", and "mu" for "micro", and the degree symbol for Celsius temperature. These three characters will be provided in 8-bit ASCII (see Part III, next month). Meanwhile, for these, and also for such equipment that has only a single case, there is a standard way of representing the SI units and prefixes. This is given in International Standard 2955, "Representations of SI Units and Other Units for Use in Systems with Limited Character Sets", and also in American Standard X3.50-1976.

To keep the record straight, let's first look at the characters used for the prefixes. They're shown in Table 2, which indicates multiples from 10 to the − 18 up to 10 to the + 18:

10 + i	i	10 − i
exa (E)	18	atto (a)
peta (P)	15	femto (f)
tera (T)	12	pico (p)
giga (G)	9	nano (n)
mega (M)	6	micro (μ)
kilo (k)	3	milli (m)
hecto (h)	2	centi (c)
deka (da)	1	deci (d)

Table 2. Metric Prefixes

Above 3 there are no powers except multiples of 3. This practice breeds better comprehension, like marking off three's in writing numbers of many digits. Also, as a memory convenience, all symbols are upper case for powers greater than + 3. And there are no conflicts with the symbols for the units of measurement.

Now, again for the record, here are the ASCII character(s) used as symbols for the units:

A	ampere	cd	candela
Bq	becquerel	d	day
C	coulomb	g	gram
°C	degree celsius	h	hour
F	farad	l	litre
Gy	gray	lm	lumen
H	henry	lx	lux
J	joule	μ	micro
K	kelvin	m	metre
N	newton	min	minute (time)
Ω	ohm	mol	mole
Pa	pascal	rad	radian
S	siemens	s	second (time)
T	tesla	sr	steradian
V	volt	t	tonne/metric ton/ megagram
W	watt		
Wb	weber		

Table 3. Metric Units

Table 3 shows the rules clearly. Units not named after people are all lower case, as shown in the righthand column (although I do know a Mr. Day). In the lefthand col-

umn are the units that are named after people. The names of the units are not capitalized at all, but the symbols begin with an upper case letter.

I said previously that there were no conflicts between unit and prefix symbols. But you've probably noticed "d" for both "day" and "deci", "h" for both "hour" and "hecto", "m" for both "metre" and "milli", and "T" for both "tesla" and "tera". OK. But there isn't any confusion in actual usage, because the prefix precedes the unit:

dd	is a deciday (2.4 hours)
hh	is a hectohour (100 hours)
hH	is a hectohenry (but don't ever use the term)
mm	is a millimetre
Mm	is a megametre (1/300 the speed of light)
TT	is a teratesla (Wow!)

I am not suggesting that the prefixes should be applied to other than the primary metric units (the second is the primary time unit; hour and day are not), even though the timesharing system I customarily use figures my time in millihours. But when you get accustomed, the prefixes are very valuable in other ways. For example, an American billion is a kilomillion, whereas the British billion is a megamillion! And my metric teaching program understands such things as kilofathoms.

The "space" character is also vital to correct SI usage. It must occur between values and units, like 123.6 mm, and 22 °C.

And don't forget another peculiarity of ASCII as an international alphabet: (1/14) is absolutely not defined as a "decimal point" (nor is it defined as "period", which in the United Kingdom is "full stop"). For most of the rest of the world, the comma (1/12) is the decimal marker, and the period is used to mark off threes. That's why the recommended practice for marking off threes is to use the space, not either comma or period. E.g., "1 234 567 mm".

To save you the bother of looking up the standards for use with limited character sets, here is the algorithm:

1. If you have ASCII with both cases of alphabet, the three missing symbols are handled as:

ohm	for Ω
Cel (initial cap)	for °C
u (lower case)	for μ (micro)

2. If you have only one case of alphabet (either upper or lower), use it, and these three replacements remain as:

OHM		ohm
CEL	or	cel
U		u

And in addition:

S (siemens)		SIE		sie
h (hour)	become	HR	or	hr
t (tonne)		TNE		tne

Examples:

16 UOHM is 16 μΩ

373.15 K = 100 Cel

Notice that no plurals are used in symbol combinations — MICROOHMS, but UOHM.

ASCII AND KEYBOARDS

Technically, a keyboard is an ASCII keyboard if it generates the proper codes for the full set of ASCII graphic and control characters. Moreover, none of the graphic characters should have any control properties.

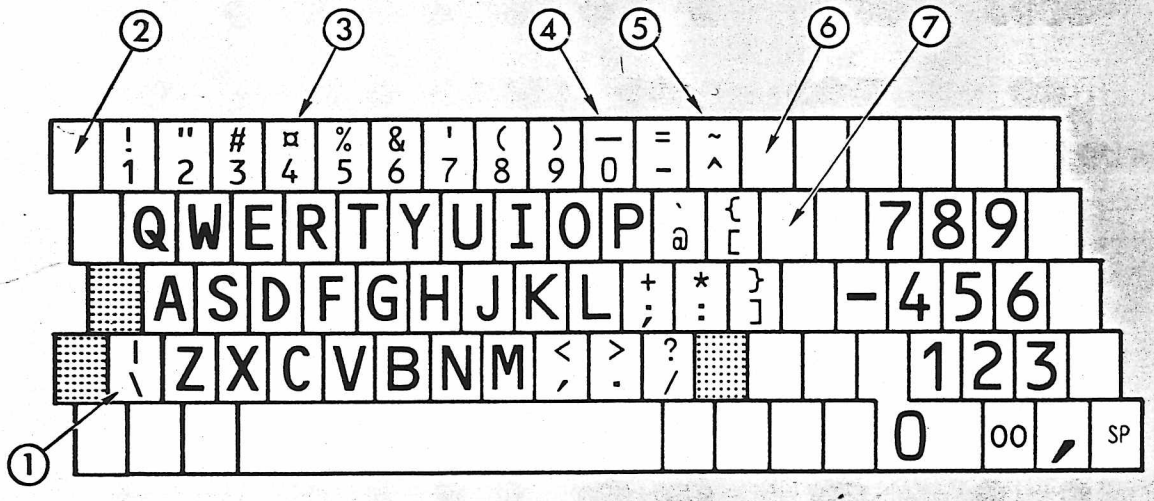
There are many types of special keyboards — Dvorak, a two-sided one used like an accordion with the hands in a vertical plane, Touch-Tone and its derivatives, etc. There are no formal standards to relate these keyboards to ASCII. For typewriter-style keyboards, however, there are two versions given in the American National Standard. One is derived from the usual electric typewriter keyboard, the other is called the "bit-paired" keyboard. Only the bit-paired keyboard will be shown and discussed here, because the other form is the subject of proposals for extensive change due to the growth of Word Processing. ANSI Committee X4A12 is studying this now.

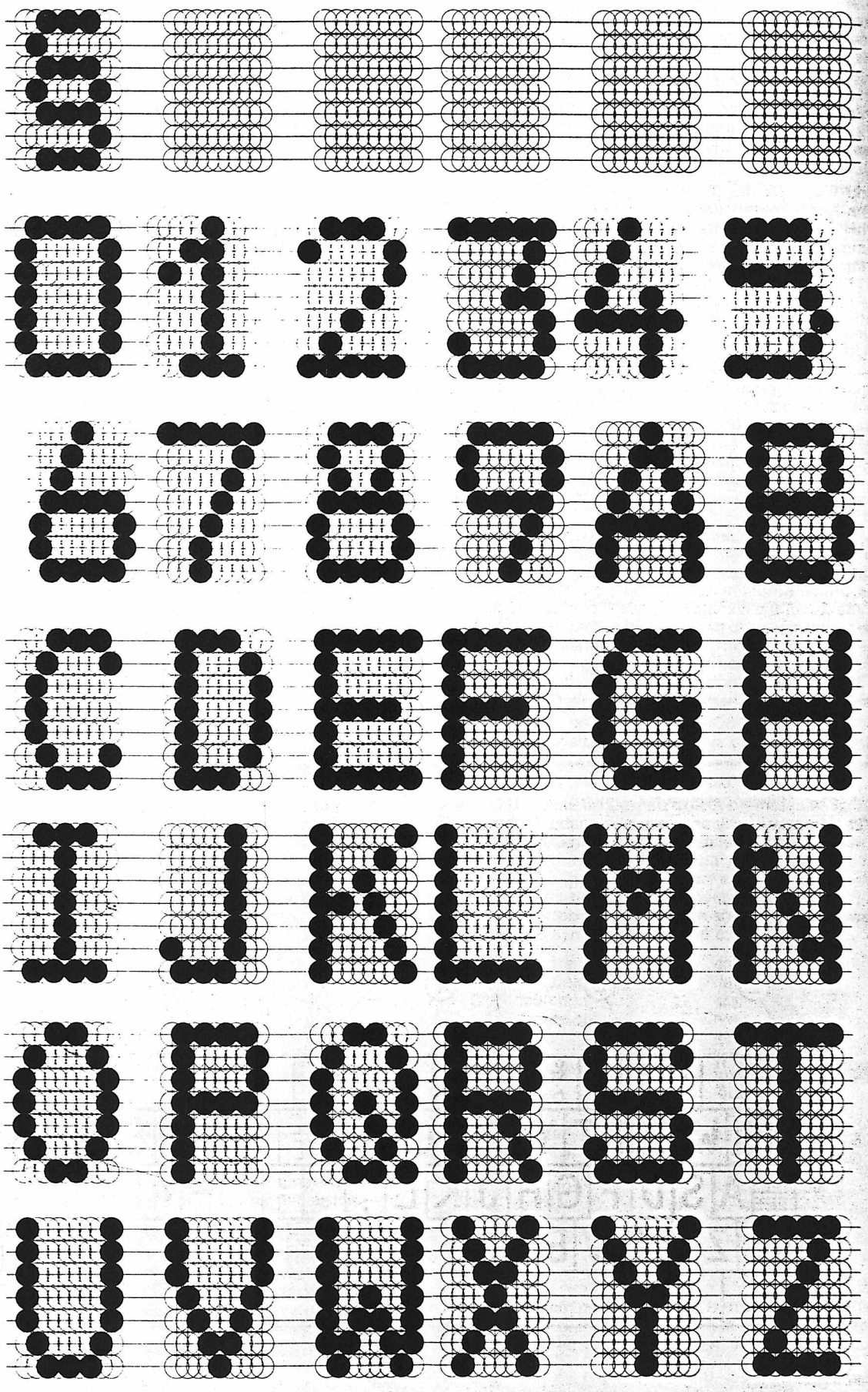
The bit-paired keyboard was designed for minimum circuitry cost. Thus the "at" symbol (4/0) is paired with the accent grave (6/0), "A" (4/1) with "a" (6/1), and "+" (2/11) with ";" (3/11). Thus the shift key affects each other key by only a 1-bit change.

This keyboard is shown in Figure 9. It is the interchange keyboard of ECMA-33. The numbered arrows key to the notes on changes that would make this ECMA keyboard into the ANSI keyboard for ASCII. It is also equivalent to the keyboard of ISO Standard 2530-1975 (Footnote 2).

Notes for Figure 9

1. *For and ANSI keyboard, this key is put to the right of the circumflex key, on the top row (see Note 6). The Shift Key is put in its place.*
2. *If this key exists and is available, the ECMA and ISO standards put the underscore here, removing it from the "zero" key.*
3. *The ANSI keyboard of course puts a "$" here in place of the international currency symbol.*
4. *This is where the underscore is removed for the 48-key keyboard (see Note 2).*





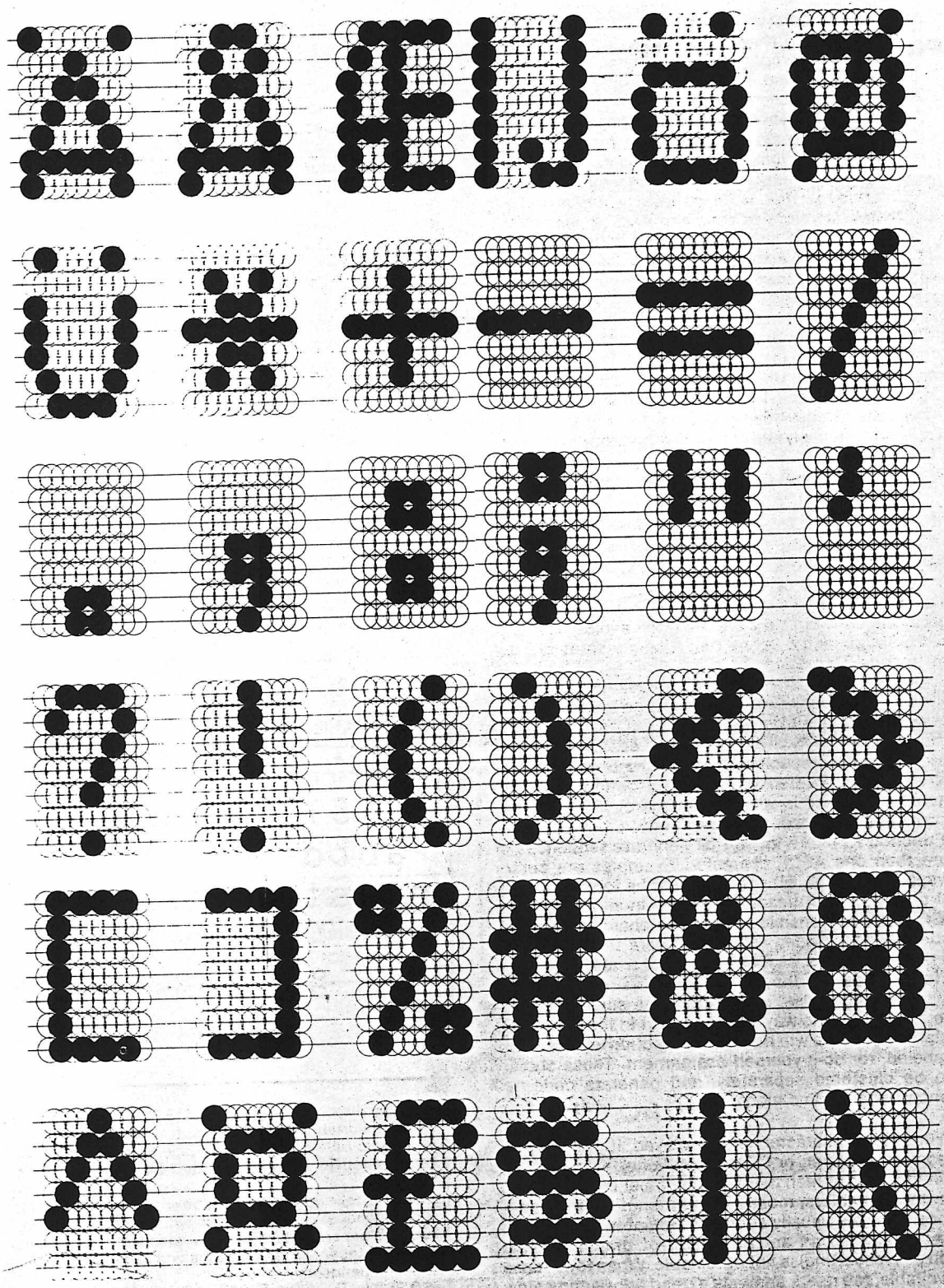


Figure 10.

5. *Here ECMA and ISO show the "overline" instead of the "tilde." It's a question of styling.*
6. *The ANSI keyboard has the reverse slash and vertical bar here, rather than between the shift key and "Z" (see Note 1).*
7. *The ANSI keyboard specifies the underscore here, in both shifts, rather than the positions shown as options in Notes 2 and 4. Practically no keyboards follow this. In fact, as I am entering this text, this is the **only** key where my Infoton Vistar deviates from the ANSI standard. It has Line Feed there, with Return to its right — a very sensible arrangement.*

Customarily, the Control Key is also tied to bit-pairing in such keyboards. The standards recommend that characters created in combination with the Control Key should use the graphic key in sticks 4 or 6 units higher. Thus "X" (5/8) or "x" (7/8) in combination with the Control Key produce CAN (1/8). Unfortunately this also means that Control-C generates ETX (0/3). And whereas Control-X as CAN is used frequently, to erase an input line of text, ETX is not often wanted. Yet it is a common miskeying to hit C rather than X. In many timesharing systems you will get a disconnect rather than a line delete.

Control and Function Keys

The so-called "QWERTY" arrangement is prevalent throughout the world. Even the French "AZERTY" set is being considered for change. But on top of these basics there are hundreds of keyboard varieties. Some of them have "dead keys" (i.e., the platen or printing element is not advanced when they are hit). This avoids having to use BS for accented letters, but it also creates difficulties in code generation.

There are some general good practices that ASCII keyboards should follow. To facilitate usage by those experienced with typewriters, all controls not used with typewriters should be located outside the customary touch-typing area. As a specific example, the Break/Interrupt key should be located where it is a definite effort to reach it (not mixed in with the keyboard). ISO 3244 may be consulted for these considerations.

Function Keys are those that generate sequences of more than one ASCII character. Examples are cursor keys, Erase-to-EOL, etc. They should be located in special clusters. Most importantly, they must all generate ASCII codes for transmission when in character-at-a-time mode. I know of video terminals where the cursors do not generate codes, as they should not while in full page buffered mode; but they still operate in line mode without generating codes. In this case the screen is alterable, but there is no way of detecting it in the computer.

Many keyboards will have some function keys that are unlabeled, for do-it-yourself assignment. These should also be clustered separately, and generate code sequences when in line mode.

ASCII AND DISPLAY PRINTING

When ASCII characters are displayed, it may be on a video screen, paper, or COM (microfiche).

On the video screen there are a number of methods to form the characters, mostly at the manufacturer's preference. They are usually at pica (constant-width) spacing for economy, so an approximation of graphic quality (such as typesetting) is not obtainable. When lower case is available, the risers and tails extend above and below the line for some screens. In others, they fall within the boundary lines of the upper case characters. They may be shown in inverse video (light background block), or highlighted by different brightness or blinking. Controls for this work will be taken up in Part III of this article.

For paper copy one usually finds either direct impact of a formed letter, or stylus printing. Either method is suitable to proportional spacing if desired. Recently there has been a general trend toward using the 7x9 dot matrix shapes of ECMA Standard 42 for stylus printers. This set of graphics is shown in Figure 10.

For hard print elements, of course, one can get a nearly infinite variety of styles and fonts. There are only two, however, specifically associated with computers — OCR-A and OCR-B. "OCR" stands for "Optical Character Recognition", meaning that the shapes are so styled that a computer-controlled scanner can read the characters as printed on paper, and encode them directly from their shapes.

OCR-A is not suitable for human reading. It's the funny looking one with the diamond-shaped letter "Oh." I won't dignify it by showing the font here. It was thought formerly, with technology of that day, that making humans work harder to read letters would make it easier and thus cheaper for computers to read them. This argument turned out to be specious, and with today's technology there is no need to use anything other than OCR-B.

OCR-B is specified in ISO 1073/2, ECMA-30, and ANSI X3.49. It is the font shown in Figure 11. I have it on my IBM golfball typewriter at home, and on my daisywheel element at the office. So it should be available for most hard elements, including the carousel type.

The first six rows correspond to ASCII sticks 2-7. In the first row, the pound and universal currency symbol are for replacement as needed. In the fourth row, the underline is discontinuous; a continuous form is shown in the auxiliary set. This set also contains a matching accent acute instead of single quote, the real circumflex (not an up arrowhead), a cedilla, and an "m" of better proportion. □

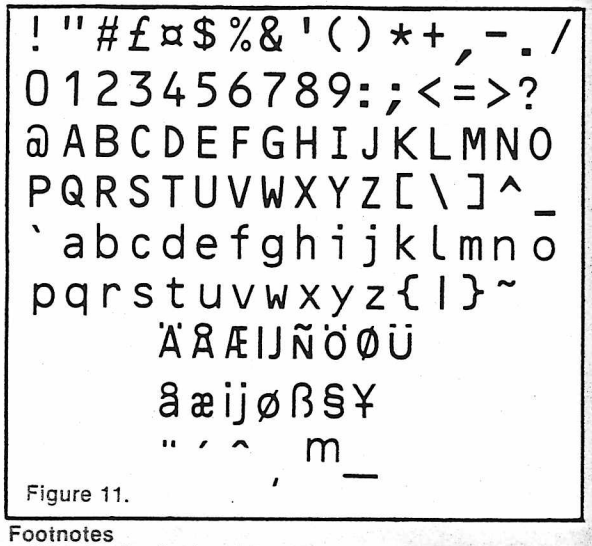


Figure 11.

Footnotes

1. With the distribution, ECMA said "ECMA-53 is an attempt to improve portability of programs. It links the language character sets defined by the language standards, their coded representatives by means of the 7-bit code and the implementations on data carriers (punched tape, punched cards, magnetic tape and magnetic tape cassettes and cartridges). It is a standard of a new type in which already standardized features are assembled in a new standardized combination aimed at supporting interchange and decreasing implementation dependency."
2. ISO 2530 is for the alphanumeric area of the keyboard only. It is augmented by ISO 3243-1975 — Keyboards for Countries whose Languages have Alphabetic Extenders, Guidelines for Harmonizations, and also by ISO 3244-1974 — Principles Governing the Positioning of Control Keys on Keyboards.
   The fact that these are "guidelines" and "principles" indicate the complexity of the subject. Typewriter manufacturers now supply over a hundred different keyboard arrangements, as their catalog will indicate.

INSIDE ASCII

CODE EXTENSION — GENERAL PRINCIPLES

Over ten years ago it was recognized that ASCII was the basis for codification of the various symbols used throughout the world. Through it, libraries could store encoded books as well as printed books. And while electronic mail may be quite simple with ASCII and its Roman alphabet, that's not the alphabet of all countries. The USSR uses Cyrillic, the Japanese use Katakana, and the Arab world uses its own semi-script alphabet. Moreover, to send a mathematics textbook by electronic mail one would have to be able to encode the formulas and special symbols peculiar to mathematics, which includes many Greek characters!

This is where the ESCape character and ESCape sequences come in. You can get the whole complicated story from ISO Standard 2022 (or ECMA-35) on Code Extension Procedures. But it will be easier to think of reproducing many ASCII Code Tables on the pages of a book, then replacing the ASCII symbols on all but the first page with the other alphabets we need.

Then we make sure that everyone in the world has the same (code) book. (The resemblance to military code books is intentional.) That's done by registering the page number assignment to characters (actually either a control set or a graphic set, but not both) with the French Standards Body AFNOR. That's the Association Francaise de Normalisation, Tour Europe Cedex 7, 92080 Paris La Defense, FRANCE. But you'll find it perhaps easier to get it from ANSI (see data in the first installment, INTERFACE AGE, May 1978).

The registration procedure is spelled out in ISO Standard 2375. It is carefully controlled to prevent frivolity and cluttering up the assignment books, for that all costs money. But the important control and graphic sets of the world may be registered and assigned their own unique ESCape sequence for calling or invoking them.

CODE EXTENSION — BASIC RULES

The control ESC, when encountered in a datastream, means that all characters following it, up to and including the first character from sticks 3 to 7, have special interpretation. The delimiting character is called a "final" (F). Those between ESC and the final are called "intermediates" (I). All of the codes in stick 2 can serve as intermediate characters in ESCape sequences of 3 or more characters in length. The entire group of characters from ESC through the final is called an ESCape sequence.

ESCape sequences obviously require buffers for interpretation, for we cannot know, when they begin, how long they will be. Sequences of length 2 are for single controls. If the character following ESC is from stick 3, the sequences are for private usage, of the class Fp. If it is from sticks 4 or 5, they mean single controls, of the class Fe, from an appropriate set of 32. If from sticks 6 or 7 (except 7/15), they are of the class Fs, composed of single controls. This is elementary extension.

A more complex type of extension is the simulation of one or more 8-bit character sets by alternating between two 7-bit sets. The home base set consists of the C0 (32 controls) set and the G0 (94 graphics plus space and DEL). The alternate sets consist of the C1 (32 controls) set and and the G1 (94 graphics plus space and DEL). The 8-bit set (it doesn't have to be just theoretical if you have a full 8-bit capability) consists of the four parts C0-G0-C1-G1.

These four types of sets are all invoked (designated) by 3-character ESCape sequences in this manner, where F is the final (3rd) character:

Sequence	Invokes Set Type
ESC 2/1 F	C0
ESC 2/2 F	C1
ESC 2/8 (or 2/12) F	G0
ESC 2/9 (or 2/15) F	G1

The final character "F" selects the particular set to invoke. Once invoked, encountering or entering an SO shifts to the G1 set in force; an SI shifts to the G0 set in force. SO and SI do not affect the control set.

ISO Standard 2022 defines these matters in far more detail, but that is enough for here. That document is complicated and ingenious, and deserves substantial study.

THE CODE EXTENSION REGISTRY

Table 1 identifies the graphic sets registered to date. Table 2 identifies the control sets registered to date. Re-

PART 3 OF 3 PARTS

By R.W. Bemer

member that these assignments, once registered, may never be changed!

Regis. No.	Final Char.	Name
002	4/0	IRV (Intl. Reference Version) Graphics
004	4/1	UK Graphics
006	4/2	US Graphics (ASCII)
008-1	4/3	NATS Main Graphic Set (Finland, Sweden)
008-2	4/4	NATS Additional Set (Finland, Sweden)
009-1	4/5	NATS Main Graphic Set (Denmark, Norway)
009-2	4/6	NATS Additional Set (Denmark, Norway)
010	4/7	Swedish Basic Graphics
011	4/8	Swedish Graphics for Names
013	4/9	JIS Katakana Graphics
014	4/10	JIS Roman Graphics
015	5/9	Italian Graphics
017	5/10	Spanish Graphics
018	5/11	Greek Graphics
019	5/12	Latin-Greek Graphics
021	4/11	German Graphics
025	5/2	French Graphics
027	5/5	Latin-Greek Mixed Graphics (Greek Capitals only)
031	5/8	Greek Alphabet Set for Bibliographic Use

For a G0 set the ESCape sequence is
ESC 2/8 plus the final shown.
For a G1 set the ESCape sequence is
ESC 2/9 plus the final shown.

Table 1. Registered Graphic Character Sets

Regis. No.	Final Char.	Name
001	4/0	ISO 646 Controls
007	4/1	Scandinavian Newspaper Controls
026	4/3	IPTC Controls

The ESCape sequence for a C0 set is
ESC 2/1 plus the final shown.

Table Vb. Registered Control Character Sets

Table 2. Registered Control Character Sets

The registry set is available from AFNOR for approximately 172 French francs, say $35. It would be vital for an equipment or software manufacturer to have it, and it comes in a beautiful 4-ring binder symbolizing worldwide interchange compatibility. But the summary provided here will fill most needs.

. . .the work I had to do to compact the standard, trying to make it understandable, turned up more than unreadability. So it's back to the drawing board, perhaps for a considerable period of time. . .it's sometimes useful to have symbols whose meaning you can reassign without harm to programming languages. . .

CONTENT OF THE EXTENDED SETS

Figure 1a shows, against the ISO Code, International Reference Version, how the other graphic sets differ in the column/row positions shown. The rows are keyed to Table 1, reminding you that ASCII is "006", or "ISO 646-006".

From this figure we can see that many countries need accented letters as individual characters, not compound via BS (BackSpace). This is particularly true for the double sets 008 and 009, for Scandinavian newspaper transmission, which have characters that cannot be made from ASCII in compound form. For example — Ring-A, a solid, and the angle open and closed quotes.

col	02				03		04	05					06	07			
row	01	02	03	04	10	15	00	11	12	13	14	15	00	11	12	13	14
002	!	"	#	¤	:	?	@	[	\	]	^	_	`	{	|	}	-
004			£	$													
006				$													~
008-1				$			ua	Ä	Ö	Å	■		ub	ä	ö	å	
009-1		«	»	$			ua	Æ	Ø	Å	■		ub	æ	ø	å	
010								Ä	Ö	Å				ä	ö	å	
011							É	Ä	Ö	Å	Ü		é	ä	ö	å	ü
014				$					¥								
015			£	$			§	°	ç	é			ù	à	ò	è	ì
017			£	$			§	¡	Ñ	¿					ñ	ç	~
018			£	$			'										
019			£	$													¨
021				$			§	Ä	Ö	Ü				ä	ö	ü	ß
025			£	$			à	°	ç	§				é	ù	è	¨
027	Ξ		Γ		Ψ	Π	Δ	Ω	Θ	Φ	Λ	Σ					

Figure 1a. Registered Graphic Character Substitution

008-2 and 009-2 are shown in Figure 1b. Here these are not exceptions from the IRV, but rather the only graphics assigned in the set. The additions are necessary to set type for newspapers throughout Scandinavia. See the Crossbar-D, Crossbar-O, the A-E ligature, and the Icelandic Thorn.

col	04			05				06			07			
row	01	04	05	00	05	11	12	01	04	05	00	05	11	12
008-2	À	Ð	É	Þ	Ü	Æ	Ø	à	ð	é	þ	ü	æ	ø
009-2	À	Ð	É	Þ	Ü	Ä	Ö	à	ð	é	þ	ü	ä	ö

Figure 1b. Registered Additional Graphic Sets

Figure 1a doesn't show Set 031 because it deviates more and is not of that much general interest. It doesn't show the Japanese Katakana set because that is completely different from the IRV. In fact, Japanese Industrial Standard C6220-1969 is an 8-bit coded set with the IRV (see Set 014 for the dollar and yen signs) in the lower (bit 8 = 0) portion, and Set 013 in the higher portion, with space reserved for future additional controls. This Set 013 is shown in Figure 2. It is shown in its high-order position, to indicate the card codes at the same time.

Figure 2 also shows the Cyrillic set of the USSR state standard GOST 13052-67, but it is not half of an 8-bit set as the Japanese do it. Rather it is another page of extensions. After SO (Shift Out) is used, the Russian register is operative. Following SI (Shift In) it is the IRV. Although this set has no registry number now, it was submitted recently by ECMA, and we expect an assignment soon. By the way, both Katakana and Cyrillic are shown in their OCR font.

Figure 3 shows the contents of the registered control sets. Set 007 serves as control set for the graphic sets 008-1,2 and 009-1,2, for Scandinavian newspaper transmission. And set 026 is the control set for the worldwide newspaper transmission, defined by the IPTC (International Press Telecommunications Council). The 18 control positions not shown, and those where there is no entry, are the same as in the International Reference Version (646-001).

These newspapers are driving composition equipment, not line printers, so they don't need VT and FF. Their set is already defined, so they don't need SO and SI. They have (properly) assigned meaning to three device controls. And they're probably not doing payroll, so they don't need the four information separators. But they do transmit, and instead of choosing their own functions and placement they have chosen to be a registered variant of the ISO Code. And all variants within this controlled and registered cluster can at least recognize each other, even if they can't print it!

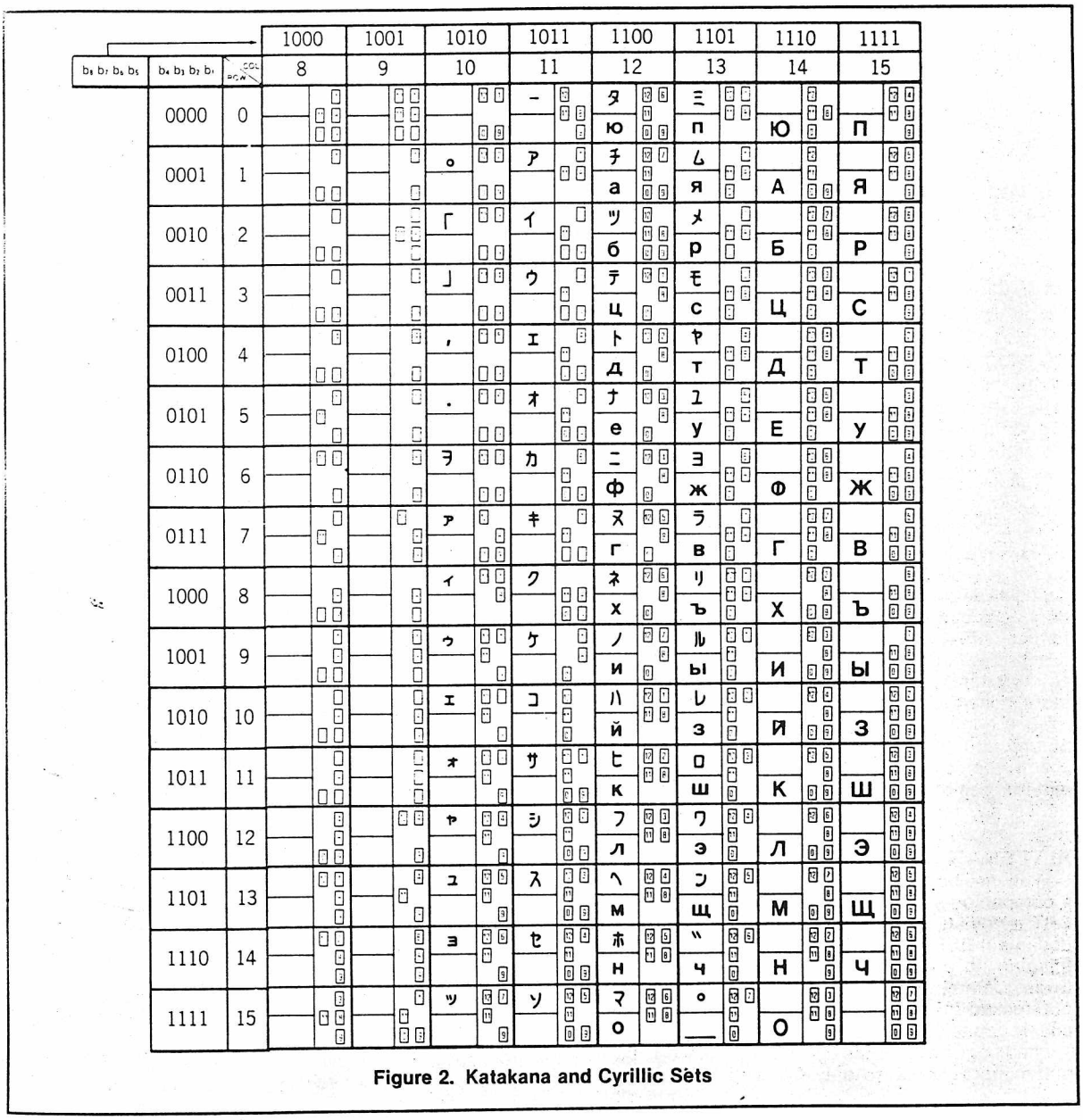


Figure 2. Katakana and Cyrillic Sets

Position	IRV 001	007	026	
0/09	HT	FO	FO	Format Control
0/11	VT	ECD	ECD	End (a typographical) CommanD
0/12	FF	SCD	SCD	Start (") CommanD
0/13	CR	QL	QL	Quad Left
0/14	SO	UR		Upper Rail
0/15	SI	LR		Lower Rail
1/01	DC1		Font 1	Change to normal
1/02	DC2		Font 2	Change to italic
1/03	DC3		Font 3	Change to bold
1/08	CAN	KW	KW	Kill Word (through previous space)
1/12	FS	SS	SS	SuperShift
1/13	GS	QC	QC	Quad Center
1/14	RS	QR	QR	Quad Right
1/15	US	JY	JY	JustifY

Figure 3. Registered Control Character Substitution

CODE EXTENSION IN ACTION

To illustrate the operation of code extension, let's imagine some equipment that may not exist now:

- A microfiche reader with automatic location controls.
- A microfiche with ASCII (the 8-bit form) on the first two pages, the other pages containing other sets such as Katakana, Cyrillic, Arabic, Greek, Hebrew, mathematical symbols, astronomical symbols, etc. Also, symbol sets for selecting typestyles, weights, rotations, sizes, and elongations.
- A display screen for the microfiche; it is touch-sensitive and generates 7-bit codes according to location touched on the display.
- As an alternative, keyboard tops with fibre optic bundles molded in as a matrix, so that the keytops can be lighted with different symbols as selected.

Now imagine that we are writing an astrology book:

- Type

```
Those of you born under the sign of Aries (
```

- Depress the "astro" key on the special keyboard
- Notice the shift in display for the fiche screen and/or the keytop lighting
- Touch the Aries symbol on the screen (or the keytop)
- Depress SI (Shift In) on the special keyboard
- And return to typing the rest of the sentence

```
) will find this month ...
```

Now imagine what a computer would do to the input stream in driving photocomposition equipment. The "astro" key generated an ESCape sequence for an astronomical graphic symbol set that would have been registered by AFNOR. When the input parser recognizes ESC, it analyzes the following characters, and then calls this set of character formation methods from the backup store, generates the character shape for Aries according to the character code after the final character, notices SI, and returns to normal mode.

Now we can envision how all of the world's printed material can be stored in machine-readable form, and interchanged recognizably!

ALTERNATE CONTROLS

Work has been in progress for several years to develop a companion standard for controls for devices such as CRT terminals. In the US this is contained in the ANSI document BSR X3.64, Additional Controls for Character Imaging. In a similar form, this C1 set is before the Codes Committee of ISO Technical Committee 97 (Computers and Information Processing) as document 2 N 868, for consideration at its 1978 May 24-26 meeting.

I had hoped to give the essence of this work in this installment. There were only two negative votes in X3, which one could presume might be answered. Unfortunately, the work I had to do to compact the standard, trying to make it understandable, turned up more than unreadability. It turned up many logical flaws and ambiguities. So it's back to the drawing board, perhaps for a considerable period of time.

Figures 4a through 4e will give, however, some flavor of the controls under consideration.

Figure 4 shows the controls of Format Type (FT) 1 and 2. Format 1 is either the single character of the 8-bit set, shown in the first column as "Ce", or the 2-character sequence of the type "ESC Fe", where Fe is a final character taken from 4/00 to 5/15, and whose column designation is 4 less than Ce. I.e., in an 8-bit code, INDex would be 8/04. In a 7-bit code it would be ESC 4/04. Format 2 is of the type "ESC Fs", where Fs is a final taken from 6/00 to 7/14.

Figures 4b through 4e show controls with formats beginning with the control "CSI", defined in Figure 4a to be either 9/11 (in the 8-bit set) or "ESC [" (in the 7-bit sets). The six possible formats are:

```
3a = CSI Pn F          4a = CSI Pn I F
3b = CSI Pn ; Pn F     4b = CSI Pn ; Pn I F
3c = CSI Ps F          4c = CSI Ps I F
```

Pn stands for numeric parameter(s), Ps for a variable number of selective parameters separated by semicolons. The type 4 formats differ from type 3 only in inserting the intermediate character 2/00 just prior to the final.

In the figures, the parameter value enclosed in parentheses is the default value. That is, if the parameters are not actually inserted, i.e., being null, then the effect is the same as if the default value(s) were inserted.

To give an example of how these controls operate, look in Figure 4d for the second mnemonic, SGR (Select Graphic Rendition). It is represented first by CSI, the Control Sequence Introducer, the parameter, and the final 6/13. This means that when the 4-character string

ESC [6 m

is encountered, it should turn on rapid blink in the field(s) specified on your video screen.

```
AL  = Active Line (containing AP)
AP  = Active Position (where the cursor is)
EF  = Editor Function
FE  = Format Effector
HT  = Horizontal Tabulation
IN  = INtroducer
PAD = Primary Auxiliary Device
RD  = Received Datastream
SAD = Secondary Auxiliary Device
SD  = String Delimiter
VT  = Vertical Tabulation
QA  = Qualified Area (defined by DAQ, SPA, EPA)
rfs = reserved for future standardization
```

Abbreviations for Figures 4a through 4e.

Fe	FT	Type	Param	Mnem	Name
8/00-03	1				(rfs)
8/04	1	FE		IND	INDex
8/05	1	FE		NEL	NExt Line
8/06	1			SSA	Start of Selected Area
8/07	1			ESA	End of Selected Area
8/08	1	FE		HTS	Horizontal Tabulation Set
8/09	1	FE		HTJ	Horiz. Tabul. with Justification
8/10	1	FE		VTS	Vertical Tabulation Set
8/11	1	FE		PLD	Partial Line Down
8/12	1	FE		PLU	Partial Line Up
8/13	1	FE		RI	Reverse Index
8/14	1	IN		SS2	Single Shift 2
8/15	1	IN		SS3	Single Shift 3
9/00	1	SD		DCS	Device Control String
9/01	1			PU1	Private Use 1
9/02	1			PU2	Private Use 2
9/03	1			STS	Set Transmit State
9/04	1			CCH	Cancel CHaracter
9/05	1			MW	Message Waiting
9/06	1			SPA	Start of Protected Area
9/07	1			EPA	End of Protected Area
9/08-10	1				(rfs)
9/11	1	IN		CSI	Control Sequence Introducer
9/12	1	SD		ST	String Terminator
9/13	1	SD		OSC	Operating System Command
9/14	1	SD		PM	Privacy Message
9/15	1	SD		APC	Application Program Command

Fs	FT	Mnem	Name
6/00	2	DMI	Disable Manual Input
6/01	2	INT	INTerrupt
6/02	2	EMI	Enable Manual Interrupt
6/03	2	RIS	Reset to Initial State

Figure 4a. Controls for Character-Imaging Devices

Final	FT	Type	Param	Mnem	Name
4/00	3a	EF	(1)	ICH	Insert CHaracter
4/01	3a	EF	(1)	CUU	CUrsor Up
4/02	3a	EF	(1)	CUD	CUrsor Down
4/03	3a	EF	(1)	CUF	CUrsor Forward
4/04	3a	EF	(1)	CUB	CUrsor Backward
4/05	3a	EF	(1)	CNL	Cursor Next Line
4/06	3a	EF	(1)	CPL	Cursor Preceding Line
4/07	3a	EF	(1)	CHA	Cursor Horizontal Absolute
4/08	3b	EF	(1;1)	CUP	CUrsor Position
4/09	3a	EF	(1)	CHT	Cursor Horizontal Tabulation
4/10	3c	EF		ED	Erase in Display
			(0)		From AP to end (inclusive)
			1		From start to AP (inclusive)
			2		All of display
4/11	3c	EF		EL	Erase in Line
			(0)		From AP to end (inclusive)
			1		From start to AP (inclusive)
			2		All of line
4/12	3a	EF	(1)	IL	Insert Line
4/13	3a	EF	(1)	DL	Delete Line
4/14	3c	EF		EF	Erase in Field
			(0)		From AP to end (inclusive)
			1		From start to AP (inclusive)
			2		All of field
4/15	3c	EF		EA	Erase in Area
			(0)		From AP to end (inclusive)
			1		From start to AP (inclusive)
			2		All of QA
5/00	3a	EF	(1)	DCH	Delete CHaracter
5/01	3c			SEM	Select editing Extent Mode
			(0)		Edit in display
			1		Edit in AL
			2		Edit in field
			3		Edit in QA
5/02	3b		(1;1)	CPR	Cursor Position Report
5/03	3a	EF	(1)	SU	Scroll Up
5/04	3a	EF	(1)	SD	Scroll Down
5/05	3a	EF	(1)	NP	Next Page
5/06	3a	EF	(1)	PP	Preceding Page
5/07	3c	EF		CTC	Cursor Tabulation Control
			(0)		Set HT stop at AP
			1		Set VT stop at AL
			2		Clear HT stop at AP
			3		Clear VT stop at AL
			4		Clear all HT stops in AL
			5		Clear all HT stops in device
			6		Clear all VT stops in device
5/08	3a	EF	(1)	ECH	Erase CHaracter
5/09	3a	EF	(1)	CVT	Cursor Vertical Tabulation
5/10	3a	EF	(1)	CBT	Cursor Backward Tabulation

Figure 4b. Controls for Character-Imaging Devices

Final	FT	Type	Param	Mnem	Name
6/00	3a	FE	(1)	HPA	Horizontal Position Absolute
6/01	3a	FE	(1)	HPR	Horizontal Position Relative
6/02	3a		(1)	REP	REPeat
6/03	3a		(0)	DA	Device Attributes
6/04	3a	FE	(1)	VPA	Vertical Position Absolute
6/05	3a	FE	(1)	VPR	Vertical Position Relative
6/06	3b	FE	(1;1)	HVP	Horiz. and Vertical Position
6/07	3c	FE		TBC	TaBulation Clear
			(0)		Clear HT stop at AP
			1		Clear VT stop at AL
			2		Clear all HT stops in AL
			3		Clear all HT stops
			4		Clear all VT stops
6/08	3c			SM	Set Mode
			1	GATM	Guarded Area Transfer Mode
			2	KAM	Keyboard Action Mode
			3	CRM	Control Representation Mode
			4	IRM	Insertion-Replacement Mode
			5	SRTM	Status Reporting Transfer Mode
			6	ERM	ERasure Mode
			7	VEM	Vertical Editing Mode
			8		(rfs)
			9		(rfs)
			10	HEM	Horizontal Editing Mode
			11	PUM	Positioning Unit Mode
			12	SRM	Send-Receive Mode
			13	FEAM	Format Effector Action Mode
			14	FETM	Format Effector Transfer Mode
			15	MATM	Multiple Area Transfer Mode
			16	TTM	Transfer Termination Mode
			17	SATM	Selected Area Transfer Mode
			18	TSM	Tabulation Stop Mode
			19	EBM	Editing Boundary Mode
			20	LNM	Line feed New Line Mode
6/09	3c			MC	Media Copy
			(0)		To PAD
			1		From PAD
			2		To SAD
			3		From SAD
			4		Turn OFF copying RD to PAD
			5		Turn ON copying RD to PAD
			6		Turn OFF copying RD to SAD
			7		Turn ON copying RD to SAD

Figure 4c. Controls for Character-Imaging Devices

Final	FT	Type	Param	Mnem	Name
6/10-11					(rfs)
6/12	3c			RM	Reset Mode
					(same parameters as SM)
6/13	3c	FE		SGR	Select Graphic Rendition
			(0)		Primary rendition
			1		Bold, or increased intensity
			2		Faint, decreased intensity, or secondary color
			3		Italic
			4		Underscore
			5		Slow blink (< 2.5/second)
			6		Rapid blink (> 2.5/second)
			7		Negative (reverse) image
			8		(rfs)
			9		(rfs)
			10		Primary Font
			11-19		1st to 9th alt. font (via FNT)
			20		Fraktur
6/14	3c			DSR	Device Status Report
			(0)		Ready, no malfunctions detected
			1		Busy - retry later
			2		Busy - DSR will notify ready
			3		Malfunction - retry
			4		Malfunction - DSR will notify ready
			5		Please report status (DSR or DSC)
			6		Please report AP via CPR
6/15	3c			DAQ	Define Area Qualification
			(0)		Accept all input
			1		Accept no input (protected); do not transmit (guarded)
			2		Accept graphics
			3		Accept numerics
			4		Accept alphabetics
			5		Right justify in area
			6		Zerofill in area
			7		HT stop at start of area (field)
			8		Accept no input (protected); permit transmit (unguarded)
			9		Spacefill in area

Figure 4d. Controls for Character-Imaging Devices

Final	FT	Type	Param	Mnem	Name
4/00	4a	EF	(1)	SL	Scroll Left
4/01	4a	EF	(1)	SR	Scroll Right
4/02	4b	FE	(100;100)	GSM	Graphic Size Modification
4/03	4a	FE		GSS	Graphic Size Selection
4/04	4b	FE	(0;0)	FNT	FoNT selection
			(0;0)		Primary font
			1;0		First alternative font
			...		...
			9;0		Ninth alternative font
4/05	4a	FE		TSS	Thin Space Specification
4/06	4c	FE		JFY	JustiFY
			(0)		Terminate all justify actions
			1		Fill action ON (text to/from other lines)
			2		Interword spacing
			3		Letter spacing
			4		Hyphenation
			5		Flush left margin
			6		Center text between margins
			7		Flush right margin
			8		Italian form (underscore last)
4/07	4b	FE		SPI	SPacing Increment
4/08	4c	FE		QUAD	Quad
			(0)		Flush left
			1		Flush left, fill with leader
			2		Center
			3		Center, fill with leader
			4		Flush right
			5		Flush right, fill with leader

Figure 4e. Controls for Character-Imaging Devices

Code	Symbol	Code	Symbol
10/00	(same as 02/00)	11/00	Large circle
10/01	Opening double quote	11/01	Dagger
10/02	Closing double quote	11/02	Superior (superscript) 2
10/03	Club suit	11/03	Superior (superscript) 3
10/04	Diamond suit	11/04	Rectangle
10/05	Heart suit	11/05	Parallel
10/06	Spade suit	11/06	Partial derivative
10/07	Closing single quote	11/07	Lower left corner, floor
10/08	Is implied by	11/08	Upper left corner, ceiling
10/09	Implies	11/09	Upper right corner
10/10	Multiply	11/10	Lower right corner
10/11	Plus or minus	11/11	Perpendicular
10/12	Nabla, or del	11/12	Less than or equal
10/13	Em dash	11/13	Not equal, other than
10/14	Radix point	11/14	Greater than or equal
10/15	Divide	11/15	Paragraph mark, pilcrow
12/00	Section mark	13/00	Capital pi
12/01	Double dagger	13/01	Capital psi
12/02	Dot bullet	13/02	Square bullet
12/03	Capital theta	13/03	Capital sigma
12/04	Capital delta	13/04	Integral
12/05	At least one exists	13/05	Capital upsilon
12/06	Capital phi	13/06	Therefore
12/07	Capital gamma	13/07	Capital omega
12/08	Upward arrow	13/08	Downward arrow
12/09	Right arrow	13/09	Left arrow
12/10	Dot product	13/10	Approximately equal
12/11	Degree	13/11	Opening angular bracket
12/12	Capital lambda	13/12	Logical AND
12/13	Register	13/13	Closing angular bracket
12/14	Copyright mark	13/14	Logical NOT
12/15	Capital xi	13/15	Infinity
14/00	Opening single quote	15/00	Small pi
14/01	Small alpha	15/01	Small psi
14/02	Small beta	15/02	Small rho
14/03	Small theta	15/03	Small sigma
14/04	Small delta	15/04	Small tau
14/05	Small epsilon	15/05	Small upsilon
14/06	Small phi	15/06	Check mark, radical mark
14/07	Small gamma	15/07	Small omega
14/08	Small eta	15/08	Small chi
14/09	Small iota	15/09	Logical universal quantifier
14/10	Identically equivalent	15/10	Small zeta
14/11	Small kappa	15/11	Cap intersection
14/12	Small lambda	15/12	Logical OR
14/13	Small mu	15/13	Cup, union
14/14	Small nu	15/14	Overbar
14/15	Small xi	15/15	(same as 7/15)

Table 3. Names of the Additional Graphics, 8-bit Set

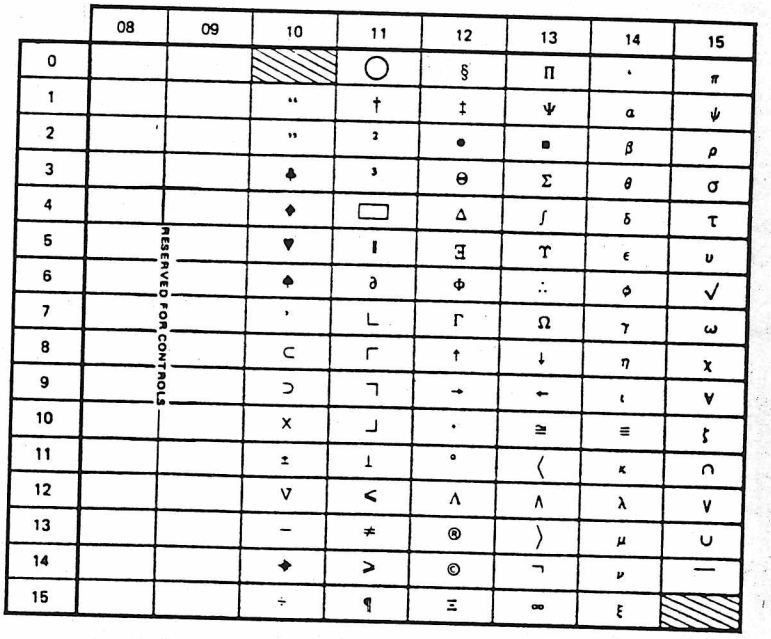


Figure 5. 8-bit ASCII Proposal

CODE EXPANSION

We have seen how ASCII was *extended* by making many related pages of the 7-bit code. It is also possible to *expand* ASCII into an 8-bit code, or even 9-bit and 10-bit if we wished, for that matter. But an 8-bit code is obviously the most logical one to concentrate on, and this has been under development for several years.

The proposed 8-bit Expanded ASCII Code is shown in Figure 5. The identification of the graphic symbols is given in Table 3.

One can observe many interesting things about this set. For example, it has the entire Greek set of small letters except for "omicron", with eleven capitals to go with others from the Roman capitals to complete the Greek set. But apparently the committee didn't follow 646-031, the Greek alphabet mentioned in Table 1. They didn't use the customary ordering "alpha-beta-gamma", the way we learn our "a-b-c's". I suppose it is argued that this set will never be used for language, only math symbols. And 646-027, shown in Figure 1a, does not demand the special capital "upsilon" shown in position 13/5. If the Greeks can agree to using a Roman capital "Y" for upsilon, could the Americans?

You'll notice some math symbols, but not enough for APL. In fact, the whole set seems highly slanted to mathematics, rather than business. Of course there are the four corner symbols for forms. Presumably the card suits will strike your eye. and you will wonder why so many other useful symbols were ignored in favor of these. Don't worry, they will always come in handy; it's sometimes useful to have symbols whose meaning you can reassign without harm to programming languages, etc. The committee were obviously bridge players, for spades collate high.

This proposal has not had real public scrutiny yet, and it must be considered no more than a proposal. Presumably X3 will agree about July that it should be sent out for formal public review and letter ballot. My guess is that it will not be adopted in just the form you see here.

FUTURE FOR ASCII

The methods are in place for codifying all symbols that people use. They may be language alphabets, signs, drawing symbols, or controls for equipments. Robots, for example. Satellites are augmenting conventional telecommunications systems. so that one can borrow cheaply and permanently from electronic libraries.

To prepare for this, other sets are being developed for registry, many through ISO Technical Committee 46/1, Automated Documentation. A 2-page mathematical symbol set is near submission, as are African sets. Work is started for Arabic, which will take about 5 sets to handle fully, although there is a commercial subset of 94 graphics. Another C1 set is being proposed for bibliographic controls. It contains four types — annotation controls, filing controls, reference controls, and subject designators. Other C1 sets can come from process control, animation and other graphics applications, etc.

West Germany has proposed a new ISO project on text communication, to harmonize teleconnection of the more than one hundred varieties of typewriters (and keyboards) throughout the world. The extension method of multiple 7-bit codes is ideal for this (8-bit codes imply too many keys or shift combinations for people to use easily.

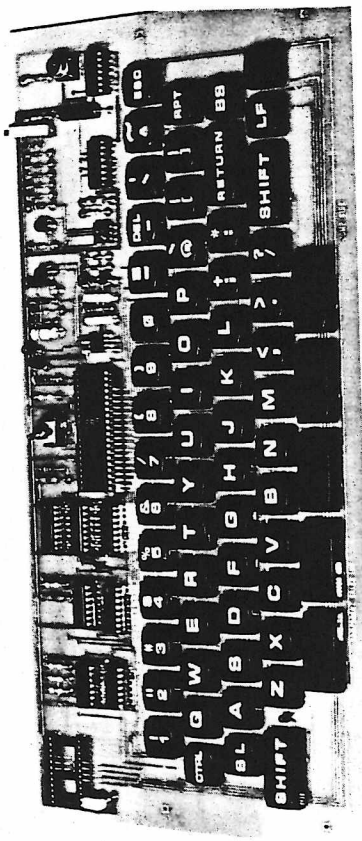
I am convinced that microcomputer users are going to develop some fantastic applications that will become widespread enough for their special graphic and control sets to be registered. How about a control set or two for sewing machines?

In fact, it is very difficult to think of any general application where one could not find a usage for these registered variants and extensions. □

Stand Alone ASCII Keyboard Specification

☆ 4 SIMULTANEOUS OUTPUTS AVAILABLE: THE ONLY ONE ON THE MARKET
1. SERIAL TTL LEVEL
2. BUFFERED 8 BIT (TRI-STATE LATCH) PARALLEL OUTPUT WITH VALID DATA SYNC PULSE AND LEVEL
3. 20 MA OPTO-ISOLATED CURRENT LOOP, POLARITY INDEPENDENT
4. EIA RS232C

☆☆ SINGLE +5 VOLT 300 MA (NOMINAL) POWER SUPPLY (REQUIRED)
☆☆☆ INDUSTRY STANDARD 2 KEY ROLLOVER ENCODER
☆☆ ANSI – COMPATIBLE KEY SET; FOR SLIM-LINE "HIDEAWAY" PACKAGING
☆☆ SEGMENTED SPACE BAR ALLOWS FAST MULTIPLE-SPACING WITHOUT REPEAT KEY
☆☆ REPEAT KEY REPEATS AT CHARACTER RATE
☆☆ USER SELECTABLE UPPER CASE ONLY (KSR/ASR/33 REPLACEMENT) OR UPPER/LOWER CASE
☆ FACTORY SET AT 110 BAUD BUT EASILY ADJUSTED BY USER TO ANY BAUD RATE FROM 110 TO 9600 BAUD
☆ FLEXIBLE PARITY
☆☆ LED INDICATOR FOR SHIFT-LOCK KEY ELIMINATES CASE UNCERTAINTY
☆☆ 24 PIN DUAL – INLINE CONNECTOR
☆☆ LOW PROFILE CASE (OPTIONAL) $40.00



$138.00 ASSEMBLED AND TESTED
Plus $3.00 handling charge. California residents add 6½% sales tax.

Orders accepted by phone or mail.
MASTERCHARGE ☆ VISA ☆ COD ☆ CHECK ☆ MONEY ORDER

COMPONENT SALES INC.
778-A BRANNAN, SAN FRANCISCO, CA 94103
[415] 861-1345

CIRCLE INQUIRY NO. 75

HONEYWELL INTEROFFICE CORRESPONDENCE

PHOENIX OPERATIONS – HONEYWELL INFORMATION SYSTEMS

DATE 1978 August 8 PHONE 8-357-2569 MAIL ZONE C61 COPIES

TO LW (Roy) Beers

FROM RW Bemer

OMPONENT Advanced Systems Engineering

SUBJECT ASCII - SOME BACKGROUND

Warning! The information here may be more than, or not what, you want to know about ASCII.

ASCII has been the American Standard Code for Information Interchange since 1963. It has been (identically) the International Standard Code since 1973. For the US Government it is also Federal Information Processing Standard 1, per signature of Lyndon B Johnson in 1968. It is Alphabet 5 for the CCITT, the ruling body for international communication. It is a state standard for the Russians and the Japanese, in conjunction with their own Cyrillic and Katakana.

Why then is it not universally used?

ORIGINS

In 1958, I was in charge of Logical System Standards for IBM. Prior to that time most IBM systems used a 6-bit code, which permitted only upper case (capital) alphabet. ALGOL was coming along, as were many other new developments demanding lower case as well. Stretch had an 8-bit character in a 64-bit word (I was at Los Alamos with Fred Brooks in 1958 on this matter).

For the 8000 (later 360) my group was developing a coalescence of the existing nine (!) different BCD codes, 8-bits wide, and compatible on all media as well as internally. Word got out, and I was invited by John Gosden (now VP Equitable Life and adviser to the President's OIP) to show the work to the British Standards Institution, which had an advanced working party on these matters. H. McGregor Ross was the chief worker. I was also asked to show it to an EIA TR committee (which was where I first met Eric Clamons).

At this time, IBM spearheaded a movement to convert the Office Equipment Manufacturers Association (a moribund operation) into BEMA. The primary purpose was to control the inevitable movement to interchange standards. I wrote the draft scope and program of work for X3.

John Auwaerter of Teletype, who developed the Models 33, 35, 37 and ink-jet printer, participated heavily in code work under ATT auspices. Basically the code ended up very much like my original proposals (I take some pride that it was for a time called the Bemer-Ross code in Europe).

IBM ACTIONS

The 360 was, as everyone knows, announced prematurely. Fred Brooks (of Brooks, Amdahl, and Blaauw--360 designers) told me that the printers and card punches were not ready in ASCII, and therefore Learson (ex-Chairman of the IBM Board) made the decision for EBCDIC--Extended BCD Interchange Code. Presumably the P1 bit was there so that the 360 could run in ASCII, but the software to do so never got implemented. Those familiar with the difficulties in getting the 360 to work will appreciate this.

To this day IBM has never released EBCDIC for standardization (which it could under the code extension and registry provisions of ISO). It is a proprietary code, and IBM states publicly that they alone reserve the right to modify it.

There are several different codes still in IBM equipment! The System 3 code, for example, is an (absolutely incompatible) permutation of the 360 EBCDIC. I have not checked System 1 yet. IBM obviously intends to handle all such conversions with hard chips, and let the competitors figure it out some way. It's by now an integral part of IBM's anti-plug-compatible strategy.

HIS HISTORY

The 600/6000/66 evolutionary line is based on a near copy of the pre-360 IBM 7094. It has a 36-bit word of 6 characters of 6 bits each (BCD).

The GE Bull M-140 (circa 1966) was designed internal ASCII. But GE cancelled it, and eventually mounted a new-line effort under Richard Bloch. He wanted to confront IBM all the way with a compatible CPU like RCA did. I worked for Bloch, but knew about the RCA efforts on the inside due to a friend of my wife's (a story for my memoirs!). Therefore I wrote a series of memos on

continuing the 600 architecture instead (they make interesting reading even now). With the help of many (Walker Dix included) this came to be. As far as I know, I was the only internal GE employee on the study teams to sell the computer business.

With the Level 64, however, Honeywell Bull began to push EBCDIC strongly. I believe that Spangle concurred. But Phoenix has never been able to implement this because of the investment in software, particularly timesharing software. In other words, we live with BCD in batch and ASCII in timesharing.

HIS NOW

The ASCII software uses a 9-bit byte, 4 to a word, even though ASCII is principally 7 bits (but going on 8 according to international work in expanding the standard). Level 6 is completely ASCII.

The product calendar apparently contains a project to make a completely ASCII version of the L66 system. I don't know if it's software only, or a hardware change as well. If hardware also, it should be a 2-step change--software first, not concurrently.

One motivation for this is that new and different software is salable and rentable, whereas the old is not. Another is networking with Level 6. This will not be an easy project. Our current GCOS III is ill-structured and overly patched already, and no one has been willing so far to spend any money to make it healthy. Code relationships are implicit in the programs, but nowhere documented. It's rewrite and retest all the way!

EXTERNAL FACTORS

Teletype now accommodates EBCDIC as well as ASCII in their terminals. They do so to make a buck in the IBM environment, as many do. ASCII is the code for "clear" traffic, but the X.25 protocol permits "private" traffic and thus EBCDIC.

IBM has stated loudly and clearly that they will use codes of many more bits than 8 for future systems. The fibre optic interface gives us this clue also. They know that EBCDIC is too ill-structured to expand, whereas ASCII is designed for just such purposes. They are prepared to move to ASCII. Such a move would jolt the add-on people terribly.

HONEYWELL INTEROFFICE CORRESPONDENCE

LW (Roy) Beers 4 1978 August 8

The microcomputer world is all ASCII. Of course HIS may never use them in its systems, but it might be wise not to preclude the option.

Bob/p

RW Bemer

pak

CF 25-1

An Annotated Bibliography on the Origins of Digital Computers

BRIAN RANDELL

101-
207

The bibliography contains nearly 750 annotated and indexed citations to papers, books, and other items relating to the origins of the modern electronic computer. The topics covered range from early digital calculating devices and mechanical automata to the first stored program computers. New entries were added up to June, 1979.

Keywords and phrases: calculating machines, sequence control mechanisms, cryptanalytic devices, punched card machinery, electromechanical and relay calculators, stored program electronic computers

CR categories: 1.2, 6.21, 6.4

Introduction

This bibliography is a revised and considerably extended version of that appearing in *The Origins of Digital Computers: Selected Papers* (Ed. B. Randell: Springer Verlag (1973)). It concentrates on the period from Babbage to the invention of the stored program electronic digital computer in the mid-1940s, but also contains items relating to earlier work on mechanical calculation and on devices incorporating sequence control mechanisms. The bibliography covers events up to about 1949 when the earliest stored program computers became operational, but no attempt has been made to cover the explosive growth in the literature on electronic computers which occurred after 1946, when a large number of computer projects were started in the United States and elsewhere based directly on the plans for the EDVAC or the Institute for Advanced Study computers. All the items listed have been inspected, sometimes rather cursorily, by the author. (Some items which are still being sought are listed in Appendix 1, as a challenge to interested readers.)

The bibliography is listed in alphabetical order by author; anonymous items are listed last. In order to facilitate use of the index, when more than one publication is listed for a given author the name of the author is followed by the year of publication in parentheses. Where there is more than one publication in one year, the year is followed by a, b, ... Although the index is fairly detailed, readers are warned to check the more general works on a given subject, as well as those explicitly listed under a particular index entry.

In the remainder of this introduction, a brief survey is attempted of some of the more interesting groups of items which are new entries in the bibliography, with special attention given to those which add significantly to the information in the set of chapter introductions in *The Origins of Digital Computers.* (The annotations with the recently added items are, in general, more detailed than those for items which were in the original bibliography.)

Brief comments are also given on some of the larger bodies of historical material which particularly merit study by historians of science. One such body of material, that arising from the so-called ENIAC patent suit, is described at some length in Appendix 2. Another large collection of technical and patent literature, covering broadly similar ground, is that donated by G. R. Stibitz to Dartmouth College Library (see Anon (1973b)). Also worthy of explicit mention is the well-cataloged and indexed collection of about 500 of K. Zuse's technical papers, letters, patents, etc. held at the Gesellschaft für Mathematik und Datenverarbeitung, Bonn. In general, however, the patent literature is not covered in the present bibliography. It is to be hoped that before long someone with the necessary specialist skills and experience will undertake the task of surveying this literature for material relating to the origins of digital computers.

General Works

In the last few years there has appeared a number of very useful books on the history of calculators and computing. The best known, at least in the English speaking world, is one from The Office of Charles and Ray Eames, edited by G. Fleck, entitled *A Computer Perspective.* Quoting from the review by D. E. Knuth in *Historia Mathematica:* "The style of presentation is aimed at a popular audience and it is a fascinating way to present history; it depicts the spirit of the times, giving isolated details rather than stating a large number of organized facts. But it is also useful as a guide for scholarly research, since the picture credits and index at the rear of the book are quite complete."

© 1979, American Federation of Information Processing Societies, Inc. Permission to copy without fee all or part of this material is granted provided that copies are not made or distributed for direct commercial advantage, the AFIPS copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the American Federation of Information Processing Societies, Inc. To copy otherwise, or to republish, requires specific permission.

Author's address: Computing Laboratory, The University of Newcastle upon Tyne, Claremont Tower, Claremont Road, Newcastle upon Tyne NE1 7RU, England.

© 1979 AFIPS 0164-1239/79/020101-107 $02.00/0

Anon. (1949d) A Calculating Machine with a "Memory." Illustrated London News (25 June 1949) pp.831-833.
Provides a number of photographs of what it terms the "Manchester Automatic Sequence-Controlled Calculating Machine," i.e., the first Manchester computer.

Anon. (1949e) Northrop Acquires Electronic "Brain". Aviation Week 29 (19 Sept. 1949) p.29.
A one-column announcement about Northrop's purchase of, and plans for using, BINAC.

186 Anon. (1949f) Thirty Years Progress. Unpublished typescript. Powers-Samas Accounting Machines [1949] 12p. (In the I.C.L. Archives.)
A non-technical chronological account, concentrating on machine developments. "The range of machines [in 1919] comprised the Tabulator, Sorter Key Punch and Slide Punch, those machines at the time were the most advanced of their kind. The Tabulator was the only card-operated Printing Machine. The punching machines were more comprehensive and efficient than any others and the new Horizontal Sorter was a great advance on anything else."

Anon. (1954) History of the Company. Unpublished typescript, Powers-Samas Company (26 Aug. 1954) 5p. (In the I.C.L. Archives.)
A chronology, giving useful information concerning when the various machines were first introduced. It states that the Powers Machines were first installed in the U.K. in 1914, at the Prudential Assurance Co., who later acquired manufacturing rights, and in 1919 set up the Accounting and Tabulating Corporation of Great Britain. Multiple Counting Sorters were introduced in 1921, the Automatic Total Attachment in 1927, the Printing Counting Sorter (developed for the British Census) in 1931, the Multiplying Punch in 1941, and the Cross Adding Punch in 1943. The company became Powers-Samas in 1929 and was taken over by Vickers from the Prudential in 1945.

Anon. (1955a) The History of Fujitsu's Computers. (In Japanese). FUJI 6,4 (1955) n.p.
A valuable series of 34 captioned photographs which provide a chronological record of the relay devices and calculators developed by Fujitsu in the period 1940-1955. These are as follows: a totalisator (1940), a totalisator and adder (1941), a binary arithmetic relay calculator, used as the basis of a matrix calculator (1942), a cryptographic decoder, developed for the Japanese Navy (1943), a statistical computer, incorporating twenty input terminals (1951), a relay computer for the stock exchange (1953), an auto-correlation calculator installed at the Seismic Research Laboratory of Tokyo University (1954), the FACOM-100 general purpose relay computer (1954), the E.T.L. Mark II general purpose relay computer (1955) and the FACOM-415A and FACOM-416A statistical calculators (1955).

Anon. (1955b) Chronological Chart of Digital Computers in Japan, compared to those in Western Countries. (In Japanese). FUJI 6,4 (1955) p.216.
Covers relay and electronic devices and computers from 1940 to 1950. The Japanese entries given consist of the Fujitsu developments listed in Anon (1955a), together with: pilot model of Tokyo University relay statistical computer (1943), Tokyo University relay computer (1947), N.E.C. relay statistical computer (1951), E.T.L. Mark I and Mark II computers (1952 and 1955), and the TAC and FUJIC electronic computers (1955). The Western entries are taken mainly from a 1953 U.S. Office of Naval Research Survey of Automatic Digital Computers.

Anon. (1956) The First Alphabetical Printing Unit. Powers-Samas Gazette (Mar./April 1956) pp.8-9.
Brief description of the invention in 1916 by Charles Foster of a device which enabled "almost certainly the first alphabetical printing in the world to be produced from any kind of punched card tabulating equipment."

Anon. (1960a) I.B.M. Computers - The story of their development. Data Processing (London) 2,2 (1960) pp.90-101.
"Towards the end of the war I.B.M. built a machine called the Automatic Sequence Controlled Calculator (sometimes referred to as the Harvard Mark I)." Also covers the S.S.E.C. (1948), the Type 603 multiplier (1946), the C.P.C., the Type 628 (1958), and the Tape Processing Machine (1950), as well as later developments.

McGraw-Hill's Compilation of DATA COMMUNICATIONS STANDARDS

Edition II

Edited by
Harold C. Folts Data Communications Standards Consultant

1982

McGraw-Hill Publications Co.
1221 Avenue of the Americas
New York, New York 10020

INTERNATIONAL TELECOMMUNICATION UNION

CCITT

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

YELLOW BOOK

VOLUME VIII – FASCICLE VIII.1

6

DATA COMMUNICATION OVER THE TELEPHONE NETWORK

RECOMMENDATIONS OF THE V SERIES

VII^TH PLENARY ASSEMBLY
GENEVA, 10–21 NOVEMBER 1980

Geneva 1981

ISBN 92-61-01151-9

SECTION 1

GENERAL

Recommendation V.1

EQUIVALENCE BETWEEN BINARY NOTATION SYMBOLS AND THE SIGNIFICANT CONDITIONS OF A TWO-CONDITION CODE

(New Delhi, 1960: amended at Geneva, 1964 and 1972)

Binary numbering expresses numbers by means of two digits normally represented by the symbols 0 and 1. Transmission channels are especially well suited to the transmission of signals by a modulation having two significant conditions (two-condition modulation). These two significant conditions are sometimes called "space" and "mark" or "start" and "stop", or they may be called condition A or condition Z [1].

It is very useful to make the two conditions of a two-condition modulation correspond to the binary digits 0 and 1. Such equivalence will facilitate the transmission of numbers resulting from binary calculation, the conversion of codes for binary numbers and of codes for decimal numbers, maintenance operations and relations between transmission personnel and the personnel in charge of data-processing machines.

At first sight, it does not seem to matter whether the symbol 0 corresponds in transmission to condition A or condition Z, the symbol 1 then corresponding to condition Z or condition A or vice versa.

In telegraphy, however, when a telegraphic communication is set up and the sending of signals is stopped (called the idle condition of the line), the signal sent over the line consists of condition Z throughout the suspension of transmission.

It is logical (and for certain VF telegraph systems also essential) to use the same rule in data transmission. During the "idle periods" of transmission, condition Z should be applied to the circuit input.

Data transmission on a circuit is often controlled by perforated tape. On perforated tapes used for telegraphy, condition Z is represented by perforation. When binary numbers are represented by means of perforations, it is customary to represent the symbol 1 by a perforation. It is therefore logical to make this symbol 1 correspond to condition Z.

For these reasons, the CCITT

unanimously declares the following view:

1 In transmitting data by two-condition code, in which the digits are formed using binary notation, the symbol 1 of the binary notation will be equivalent to condition Z of the modulation, and the symbol 0 of the binary notation will be equivalent to condition A of the modulation.

2 During periods when there is no signal sent to the input of the circuit, the circuit input condition is condition Z.

3 If perforation is used, one perforation corresponds to one unit interval under condition Z.

4 In accordance with Recommendation R.31, the sending of symbol 1 (condition Z) corresponds to the tone being sent on a channel using amplitude modulation.

5 In accordance with Recommendation R.35, when frequency modulation is used, the sending of symbol 0 corresponds to the higher frequency, while the sending of symbol 1 corresponds to the lower frequency.

6 a) For phase modulation with reference phase:

the symbol 1 corresponds to a phase equal to the reference phase;

the symbol 0 corresponds to a phase opposed to the reference phase.

b) For differential two-phase modulation where the alternative phase changes are 0 degree or 180 degrees:

the symbol 1 corresponds to a phase inversion from the previous element;

the symbol 0 corresponds to a no-phase inversion from the previous element.

7 A summary of equivalence is shown in Table 1/V.1.

TABLE 1/V.1

Summary of equivalence (see Note 1)

	Digit 0 "Start" signal in start-stop code Line available condition in telex switching "Space" element of start-stop code Condition A	Digit 1 "Stop" signal in start-stop code Line idle condition in telex switching (Note 2) "Mark" element of start-stop code Condition Z
Amplitude modulation	Tone-off	Tone-on
Frequency modulation	High frequency	Low frequency
Phase modulation with reference phase	Opposite phase to the reference phase	Reference phase
Differential two-phase modulation where the alternative phase changes are 0 degree or 180 degrees	No phase inversion	Inversion of the phase
Perforations	No perforation	Perforation

Note 1 – The standardization described in this Recommendation is general, whether over telegraph-type circuits or over circuits of the telephone type, making use of electromechanical or electronic devices.

Note 2 – It primarily applies to anisochronous use.

Reference

[1] CCITT Definition: *Position A; position Z*, Vol. X, Fascicle X.1 (Terms and Definitions).

INTERNATIONAL ALPHABET No. 5

(Mar del Plata, 1968; amended at Geneva, 1972)

Introduction

A seven-unit alphabet capable of meeting the requirements of private users on leased circuits and of users of data transmission by means of connections set up by switching on the general telephone network or on telegraph networks has been established jointly by the CCITT and the International Organization for Standardization (ISO).

This alphabet – Alphabet No. 5 – is not intended to replace Alphabet No. 2. It is a supplementary alphabet for the use of those who might not be satisfied with the more limited possibilities of Alphabet No. 2. In such cases it is considered as the alphabet to be used as common basic language for data transmission and for elaborated message systems.

Alphabet No. 5 does not exclude the use of any other alphabet that might be better adapted to special needs.

1 Scope and field of application

1.1 This Recommendation contains a set of 128 characters (control characters and graphic characters such as letters, digits and symbols) with their coded representation. Most of these characters are mandatory and unchangeable, but provision is made for some flexibility to accommodate special national and other requirements.

1.2 The need for graphics and controls in data processing and in data transmission has been taken into account in determining this character set.

1.3 This Recommendation consists of a general table with a number of options, notes, a legend and explanatory notes. It also contains a specific International Reference Version, guidance on the exercise of the options to define specific national versions and application oriented versions.

1.4 This character set is primarily intended for the interchange of information within message transmission systems and between data processing systems and associated equipment.

1.5 This character set is applicable to all Latin alphabets.

1.6 The character set includes facilities for extension where its 128 characters are insufficient for particular applications.

1.7 The definition of some control characters in this Recommendation assumes that data associated with them is to be processed serially in a forward direction. Their effect when included in strings of data which are processed other than serially in a forward direction or included in data formatted for fixed record processing may have undesirable effects or may require additional special treatment to ensure that the control characters have their desired effect.

2 Implementation

2.1 This set should be regarded as a basic alphabet in an abstract sense. Its practical use requires definitions of its implementation in various media. For example, this could include punched tapes, punched cards, magnetic tapes and transmission channels, thus permitting interchange of data to take place either indirectly by means of an intermediate recording in a physical medium, or by local electrical connection of various units (such as input and output devices and computers) or by means of data transmission equipment.

2.2 The implementation of this coded character set in physical media and for transmission, taking also into account the need for error checking, is the subject of ISO publications.

TABLE I/V.3

Basic code

b_4	b_3	b_2	b_1		b_7 0	0	0	0	1	1	1	1
					b_6 0	0	1	1	0	0	1	1
					b_5 0	1	0	1	0	1	0	1
					0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	TC$_7$ (DLE)	SP	0	@	P	` ④	p
0	0	0	1	1	TC$_1$ (SOH)	DC$_1$	!	1	A	Q	a	q
0	0	1	0	2	TC$_2$ (STX)	DC$_2$	" ⑤	2	B	R	b	r
0	0	1	1	3	TC$_3$ (ETX)	DC$_3$	£(#) ②	3	C	S	c	s
0	1	0	0	4	TC$_4$ (EOT)	DC$_4$	$(¤) ②	4	D	T	d	t
0	1	0	1	5	TC$_5$ (ENQ)	TC$_8$ (NAK)	%	5	E	U	e	u
0	1	1	0	6	TC$_6$ (ACK)	TC$_9$ (SYN)	&	6	F	V	f	v
0	1	1	1	7	BEL	TC$_{10}$ (ETB)	' ⑥	7	G	W	g	w
1	0	0	0	8	FE$_0$ (BS)	CAN	(	8	H	X	h	x
1	0	0	1	9	FE$_1$ (HT)	EM	)	9	I	Y	i	y
1	0	1	0	10	FE$_2$ (LF) ①	SUB	*	:	J	Z	j	z
1	0	1	1	11	FE$_3$ (VT) ①	ESC	+	;	K	[③	k	{ ③
1	1	0	0	12	FE$_4$ (FF) ①	IS$_4$ (FS)	, ⑥	<	L	\ ③	l	| ③
1	1	0	1	13	FE$_5$ (CR) ①	IS$_3$ (GS)	-	=	M	] ③	m	} ③
1	1	1	0	14	SO	IS$_2$ (RS)	.	>	N	^ ③④	n	~ ③④
1	1	1	1	15	SI	IS$_1$ (US)	/	?	O	_	o	DEL

CCITT - 43330

14

Notes about Table 1/V.3:

Note 1 – The Format Effectors are intended for equipment in which horizontal and vertical movements are effected separately. If equipment requires the action of CARRIAGE RETURN to be combined with a vertical movement, the Format Effector for that vertical movement may be used to effect the combined movement. For example, if NEW LINE (symbol NL, equivalent to CR + LF) is required, FE_2 shall be used to represent it. This substitution requires agreement between the sender and the recipient of the data.

The use of these combined functions may be restricted for international transmission on general switched telecommunication networks (telegraph and telephone networks).

Note 2 – The symbol £ is assigned to position 2/3 and the symbol $ is assigned to position 2/4. In a situation where there is no requirement for the symbol £ the symbol # (number sign) may be used in position 2/3. Where there is no requirement for the symbol $ the symbol ¤ (currency sign) may be used in position 2/4. The chosen allocations for symbols to these positions for international information interchange should be agreed between the interested parties. It should be noted that, unless otherwise agreed between sender and recipient, the symbols £, $ or ¤ do not designate the currency of a specific country.

Note 3 – National use positions. The allocation of characters to these positions lies within the responsibility of national standardization bodies. These positions are primarily intended for alphabet extensions. If they are not required for that purpose, they may be used for symbols.

Note 4 – Positions 5/14, 6/0 and 7/14 are provided for the symbols UPWARD ARROW HEAD, GRAVE ACCENT and OVERLINE. However, these positions may be used for other graphical characters when it is necessary to have 8, 9 or 10 positions for national use.

Note 5 – Position 7/14 is used for the graphic character ‾ (OVERLINE), the graphical representation of which may vary according to national use to represent ~ (TILDE) or another diacritical sign provided that there is no risk of confusion with another graphic character included in the table.

Note 6 – The graphic characters in positions 2/2, 2/7, 2/12 and 5/14 have respectively the significance of QUOTATION MARK, APOSTROPHE, COMMA and UPWARD ARROW HEAD; however, these characters take on the significance of the diacritical signs DIAERESIS, ACUTE ACCENT, CEDILLA and CIRCUMFLEX ACCENT when they are preceded or followed by the BACKSPACE character (0/8).

4 Legend

4.1 *Control characters*

Abbreviation	Note of Table 1/V.3	Meaning	Position in the code table
ACK		Acknowledge	0/6
BEL		Bell	0/7
BS		Backspace	0/8
CAN		Cancel	1/8
CR	1	Carriage return	0/13
DC		Device control	–
DEL		Delete	7/15
DLE		Data link escape	1/0
EM		End of medium	1/9
ENQ		Enquiry	0/5
EOT		End of transmission	0/4
ESC		Escape	1/11
ETB		End of transmission block	1/7
ETX		End of text	0/3
FE		Format effector	–
FF	1	Form feed	0/12
FS		File separator	1/12
GS		Group separator	1/13
HT		Horizontal tabulation	0/9
IS		Information separator	–
LF	1	Line feed	0/10
NAK		Negative acknowledge	1/5
NUL		Null	0/0
RS		Record separator	1/14
SI		Shift-in	0/15
SO		Shift-out	0/14
SOH		Start of heading	0/1
SP		Space (see § 7.2)	2/0
STX		Start of text	0/2
SUB		Substitute character	1/10
SYN		Synchronous idle	1/6
TC		Transmission control	–
US		Unit separator	1/15
VT	1	Vertical tabulation	0/11

4.2 *Graphic characters*

Graphic	Notes of Table 1/V.3	Name	Position in the code table
(space)		Space (see § 7.2)	2/0
!		Exclamation mark	2/1
"	6	Quotation mark, Diaeresis	2/2
£	2	Pound sign	2/3
#	2	Number sign	2/3
$	2	Dollar sign	2/4
¤	2	Currency sign	2/4
%		Percent sign	2/5
&		Ampersand	2/6
'	6	Apostrophe, Acute accent	2/7
(		Left parenthesis	2/8
)		Right parenthesis	2/9
*		Asterisk	2/10
+		Plus sign	2/11
,	6	Comma, Cedilla	2/12
-		Hyphen, Minus sign	2/13
.		Full stop (period)	2/14
/		Solidus	2/15
:		Colon	3/10
;		Semi-colon	3/11
<		Less-than sign	3/12
=		Equal sign	3/13
>		Greater-than sign	3/14
?		Question mark	3/15
^	4, 6	Upward arrow head, Circumflex accent	5/14
_		Underline	5/15
`	4	Grave accent	6/0
~	4, 5	Overline, Tilde	7/14

5 Explanatory notes

5.1 *Numbering of the positions in Table 1/V.3*

Within any one character the bits are identified by b_7, $b_6 \ldots b_1$, where b_7 is the highest order, or most significant bit, and b_1 is the lowest order, or least significant bit. If desired these may be given a numerical significance in the binary system, thus:

Bit identification:	b_7	b_6	b_5	b_4	b_3	b_2	b_1
Significance:	64	32	16	8	4	2	1

In the table the columns and rows are identified by numbers written in binary and decimal notations.

Any one position in the table may be identified either by its bit pattern, or by its column and row numbers. For instance, the position containing the digit 1 may be identified:

- by its bit pattern in order of decreasing significance, e.g. 011 0001 [1];
- by its column and row numbers, e.g. 3/1.

The column number is derived from bits b_7, b_6 and b_5 giving them weights of 4, 2 and 1 respectively. The row number is derived from bits b_4,.b_3, b_2 and b_1 giving them weights of 8, 4, 2 and 1 respectively.

[1] Order of transmitting bits is not necessarily the same as shown here. For the order of the transmission of bits, see I in Recommendation V.4 or X.4 [1].

5.2 *Diacritical signs*

In the character set, some printing symbols may be designed to permit their use for the composition of accented letters when necessary for general interchange of information. A sequence of three characters, comprising a letter, BACKSPACE and one of these symbols, is needed for this composition, and the symbol is then regarded as a diacritical sign. It should be noted that these symbols take on their diacritical significance when they are preceded or followed by one BACKSPACE character; for example, the symbol corresponding to the code combination 2/7 normally has the significance of APOSTROPHE, but becomes the diacritical sign ACUTE ACCENT when it precedes or follows a BACKSPACE character.

In order to increase efficiency, it is possible to introduce accented letters (as single characters) in the positions marked by Note 3 in Table 1/V.3. According to national requirements, these positions may contain special diacritical signs.

5.3 *Names, meanings and fonts of graphic characters*

This Recommendation assigns at least one name to denote each of the graphic characters displayed in Tables 1/V.3 and 2/V.3. The names chosen to denote graphic characters are intended to reflect their customary meanings. However, this Recommendation does not define and does not restrict the meanings of graphic characters. Nor does it specify a particular style or font design for the graphic characters.

Under the provision of Note 3 of Table 1/V.3, graphic characters which are different from the characters of the international reference version may be assigned to the national use positions. When such assignments are made, the graphic characters should have distinct forms and be given distinctive names which are not in conflict with any of the forms or the names of any of the graphic characters in the international reference version.

5.4 *Uniqueness of character allocation*

A character allocated to a position in Table 1/V.3 may not be placed elsewhere in the table. For example, in the case of position 2/3 the character not used cannot be placed elsewhere. In particular the POUND sign (£) can never be represented by the bit combination of position 2/4.

6 Versions of Table 1/V.3

6.1 *General*

6.1.1 In order to use Table 1/V.3 for information interchange, it is necessary to exercise the options left open, i.e. those affected by Notes 2 to 5. A single character must be allocated to each of the positions for which this freedom exists or it must be declared to be unused. A code table completed in this way is called a *version.*

6.1.2 The Notes to Table 1/V.3, the Explanatory Notes and the Legend apply in full to any version.

6.2 *International reference version*

This version is available for use when there is no requirement to use a national or an application-oriented version. In international information processing interchange the international reference version (Table 2/V.3) is assumed unless a particular agreement exists between sender and recipient of the data.

TABLE 2/V.3

International reference version

b_4	b_3	b_2	b_1		b_7 0	0	0	0	1	1	1	1
					b_6 0	0	1	1	0	0	1	1
					b_5 0	1	0	1	0	1	0	1
					0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	TC_7 (DLE)	SP	0	@	P	`	p
0	0	0	1	1	TC_1 (SOH)	DC_1	!	1	A	Q	a	q
0	0	1	0	2	TC_2 (STX)	DC_2	"	2	B	R	b	r
0	0	1	1	3	TC_3 (ETX)	DC_3	#	3	C	S	c	s
0	1	0	0	4	TC_4 (EOT)	DC_4	¤	4	D	T	d	t
0	1	0	1	5	TC_5 (ENQ)	TC_8 (NAK)	%	5	E	U	e	u
0	1	1	0	6	TC_6 (ACK)	TC_9 (SYN)	&	6	F	V	f	v
0	1	1	1	7	BEL	TC_{10} (ETB)	'	7	G	W	g	w
1	0	0	0	8	FE_0 (BS)	CAN	(	8	H	X	h	x
1	0	0	1	9	FE_1 (HT)	EM	)	9	I	Y	i	y
1	0	1	0	10	FE_2 (LF)	SUB	*	:	J	Z	j	z
1	0	1	1	11	FE_3 (VT)	ESC	+	;	K	[	k	{
1	1	0	0	12	FE_4 (FF)	IS_4 (FS)	,	<	L	\	l	|
1	1	0	1	13	FE_5 (CR)	IS_3 (GS)	-	=	M	]	m	}
1	1	1	0	14	SO	IS_2 (RS)	.	>	N	^	n	~
1	1	1	1	15	SI	IS_1 (US)	/	?	O	_	o	DEL

CCITT-11540

18

The following characters are allocated to the optional positions of Table 1/V.3:

#	Number sign	2/3
¤	Currency sign	2/4
@	Commercial at	4/0
[	Left square bracket	5/11
\	Reverse solidus	5/12
]	Right square bracket	5/13
{	Left curly bracket	7/11
|	Vertical line	7/12
}	Right curly bracket	7/13

CCITT-43340

It should be noted that no substitution is allowed when using the international reference version.

6.3 *National versions*

6.3.1 The responsibility for defining national versions lies with the national standardization bodies. These bodies shall exercise the options available and make the required selection.

6.3.2 If so required, more than one national version can be defined within a country. The different versions shall be separately identified. In particular when for a given national use position, e.g. 5/12 or 6/0, alternative characters are required, two different versions shall be identified, even if they differ only by this single character.

6.3.3 If there is in a country no special demand for specific characters, it is strongly recommended that the characters of the international reference version be allocated to the same national use positions.

6.4 *Application-oriented versions*

Within national or international industries, organizations or professional groups, application-oriented versions can be used. They require precise agreement among the parties concerned, who will have to exercise the options available and to make the required selection.

7 Functional characteristics related to control characters

Some definitions given below are stated in general terms and more explicit definitions of use may be needed for specific implementation of the code table on recording media or on transmission channels. These more explicit definitions and the use of these characters are the subject of ISO publications.

7.1 *General designations and control characters*

The general designation of control characters involves a specific class name followed by a subscript number.

They are defined as follows:

TC – *Transmission control characters*

Control characters intended to control or facilitate transmission of information over telecommunication networks.

The use of the TC characters on the general telecommunication networks is the subject of ISO publications.

The transmission control characters are:

ACK, DLE, ENQ, EOT, ETB, ETX, NAK, SOH, STX and SYN.

FE – *Format effectors*

Control characters mainly intended for the control of the layout and positioning of information on printing and/or display devices. In the definitions of specific format effectors, any reference to printing devices should be interpreted as including display devices. The definitions of format effectors use the following concept:

a) a page is composed of a number of lines of characters;

b) the characters forming a line occupy a number of positions called character positions;

c) the active position is that character position in which the character about to be processed would appear if it were to be printed. The active position normally advances one character position at a time.

The format effector characters are:

BS, CR, FF, HT, LF and VT (see also Note 1 to Table 1/V.3).

DC – *Device control characters*

Control characters for the control of a local or remote ancillary device (or devices) connected to a data processing and/or telecommunication system. These control characters are not intended to control telecommunication systems; this should be achieved by the use of TCs.

Certain preferred uses of the individual DCs are given in § 7.2 below.

IS – *Information separators*

Control characters that are used to separate and qualify data logically. There are four such characters. They may be used either in hierarchical order or non-hierarchically; in the latter case their specific meanings depend on their applications.

When they are used hierarchically, the ascending order is:

US, RS, GS, FS.

In this case data normally delimited by a particular separator cannot be split by a higher order separator but will be considered as delimited by any higher order separator.

7.2 *Specific control characters*

Individual members of the classes of controls are sometimes referred to by their abbreviated class name and a subscript number (e.g. TC_5) and sometimes by a specific name indicative of their use (e.g. ENQ).

Different but related meanings may be associated with some of the control characters but in an interchange of data this normally requires agreement between the sender and the recipient.

ACK – *Acknowledge*

A transmission control character transmitted by a receiver as an affirmative response to the sender.

BEL – *Bell*

A control character that is used when there is a need to call for attention; it may control alarm or attention devices.

BS – *Backspace*

A format effector which moves the active position one character position backwards on the same line.

CAN – *Cancel*

A character, or the first character of a sequence, indicating that the data preceding it is in error. As a result this data is to be ignored. The specific meaning of this character must be defined for each application and/or between sender and recipient.

CR – *Carriage return*

A format effector which moves the active position to the first character position on the same line.

Device controls

DC1 – A device control character which is primarily intended for turning on or starting an ancillary device. If it is not required for this purpose, it may be used to restore a device to the basic mode of operation (see also DC2 and DC3), or for any other device control function not provided by other DCs.

DC2 – A device control character which is primarily intended for turning on or starting an ancillary device. If it is not required for this purpose, it may be used to set a device to a special mode of operation (in which case DC1 is used to restore the device to the basic mode), or for any other device control function not provided by other DCs.

DC3 – A device control character which is primarily intended for turning off or stopping an ancillary device. This function may be a secondary level stop, e.g. wait, pause, stand-by or halt (in which case DC1 is used to restore normal operation). If it is not required for this purpose, it may be used for any other device control function not provided by other DCs.

DC4 – A device control character which is primarily intended for turning off, stopping or interrupting an ancillary device. If it is not required for this purpose, it may be used for any other device control function not provided by other DCs.

Examples of use of the device controls:

1) One switching

	on – DC2	off – DC4

2) Two independent switchings

First one	on – DC2	off – DC4
Second one	on – DC1	off – DC3

3) Two dependent switchings

General	on – DC2	off – DC4
Particular	on – DC1	off – DC3

4) Input and output switching

Output	on – DC2	off – DC4
Input	on – DC1	off – DC3

DEL – *Delete*

A character used primarily to erase or obliterate an erroneous or unwanted character in punched tape. DEL characters may also serve to accomplish media-fill or time-fill. They may be inserted into or removed from a stream of data without affecting the information content of that stream, but then the addition or removal of these characters may affect the information layout and/or the control of equipment.

DLE – *Data link escape*

A transmission control character which will change the meaning of a limited number of contiguously following characters. It is used exclusively to provide supplementary data transmission control functions. Only graphic characters and transmission control characters can be used in DLE sequences.

EM – *End of medium*

A control character that may be used to identify the physical end of a medium, or the end of the used portion of a medium, or the end of the wanted portion of data recorded on a medium. The position of this character does not necessarily correspond to the physical end of the medium.

ENQ – *Enquiry*

A transmission control character used as a request for a response from a remote station – the response may include station identification and/or station status. When a "Who are you?" function is required on the general switched transmission network, the first use of ENQ after the connection is established shall have the meaning "Who are you?" (station identification). Subsequent use of ENQ may, or may not, include the function "Who are you?", as determined by agreement.

EOT – *End of transmission*

A transmission control character used to indicate the conclusion of the transmission of one or more texts.

ESC – *Escape*

22

A control character which is used to provide an additional control function. It alters the meaning of a limited number of contiguously following bit combinations which constitute the escape sequence.

Escape sequences are used to obtain additional control functions which may provide among other things graphic sets outside the standard set. Such control functions must not be used as additional transmission controls.

The use of the character ESC and of the escape sequences in conjunction with code extension techniques is the subject of an ISO Standard.

ETB – *End of transmission block*

A transmission control character used to indicate the end of a transmission block of data where data is divided into such blocks for transmission purposes.

ETX – *End of text*

A transmission control character which terminates a text.

FF – *Form feed*

A format effector which advances the active position to the same character position on a predetermined line of the next form or page.

HT – *Horizontal tabulation*

A format effector which advances the active position to the next predetermined character position on the same line.

IS_1 (US) – A control character used to separate and qualify data logically; its specific meaning has to be defined for each application. If this character is used in hierarchical order as specified in the general definition of IS, it delimits a data item called a UNIT.

IS_2 (RS) – A control character used to separate and qualify data logically; its specific meaning has to be defined for each application. If this character is used in hierarchical order as specified in the general definition of IS, it delimits a data item called a RECORD.

IS_3 (GS) – A control character used to separate and qualify data logically; its specific meaning has to be defined for each application. If this character is used in hierarchical order as specified in the general definition of IS, it delimits a data item called a GROUP.

IS_4 (FS) – A control character used to separate and qualify data logically; its specific meaning has to be defined for each application. If this character is used in hierarchical order as specified in the general definition of IS, it delimits a data item called a FILE.

LF – *Line feed*

A format effector which advances the active position to the same character position of the next line.

NAK – *Negative acknowledge*

A transmission control character transmitted by a receiver as a negative response to the sender.

NUL – *Null*

A control character used to accomplish media-fill or time-fill. NUL characters may be inserted into or removed from a stream of data without affecting the information content of that stream, but then the addition or removal of these characters may affect the information layout and/or the control of equipment.

SI – *Shift-in*

A control character which is used in conjunction with SHIFT-OUT and ESCAPE to extend the graphic character set of the code. It may reinstate the standard meanings of the bit combinations which follow it. The effect of this character when using code extension techniques is described in an ISO Standard.

SO – *Shift-out*

A control character which is used in conjunction with SHIFT-IN and ESCAPE to extend the graphic character set of the code. It may alter the meaning of the bit combinations of columns 2 to 7 which follow it until a SHIFT-IN character is reached. However, the characters SPACE (2/0) and DELETE (7/15) are unaffected by SHIFT-OUT. The effect of this character when using code extension techniques is described in an ISO Standard.

SOH – *Start of heading*

A transmission control character used as the first character of a heading of an information message.

SP – *Space*

A character which advances the active position one character position on the same line. This character is also regarded as a non-printing graphic.

STX – *Start of text*

A transmission control character which precedes a text and which is used to terminate a heading.

SUB – *Substitute character*

A control character used in the place of a character that has been found to be invalid or in error. SUB is intended to be introduced by automatic means.

SYN – *Synchronous idle*

A transmission control character used by a synchronous transmission system in the absence of any other character (idle condition) to provide a signal from which synchronism may be achieved or retained between data terminal equipment.

VT – *Vertical tabulation*

A format effector which advances the active position to the same character position on the next predetermined line.

Reference

[1] CCITT Recommendation *General structure of signals of International Alphabet No. 5 code for data transmission over public data networks*, Vol. VIII, Fascicle VIII.2, Rec. X.4.

Recommendation V.4

GENERAL STRUCTURE OF SIGNALS OF INTERNATIONAL ALPHABET No. 5 CODE FOR DATA TRANSMISSION OVER PUBLIC TELEPHONE NETWORKS [1]

(Mar del Plata, 1968; amended at Geneva, 1976 and 1980)

The CCITT,

I. *considering, firstly,*

the agreement between the International Organization for Standardization (ISO) and the CCITT on the main characteristics of a seven-unit alphabet (International Alphabet No. 5) to be used for data transmission and for telecommunications requirements that cannot be met by the existing five-unit International Alphabet No. 2;

the interest, both to the users and to the telecommunication services, of an agreement concerning the chronological order of transmission of bits in serial working;

declares the view

that the agreed rank number of the unit in the alphabetical table of combinations should correspond to the chronological order of transmission in serial working on telecommunication circuits;

---

[1] See Recommendation X.4 [1] for data transmission over public data networks.

that, when this rank in the combination represents the order of the bit in binary numbering, the bits should be transmitted in serial working with the low order bit first;

that the numerical meaning corresponding to each information unit considered in isolation is that of the digit:

0 for a unit corresponding to condition A (travail = space), and

1 for a unit corresponding to condition Z (repos = mark),

in accordance with the definitions of these conditions for a two-condition transmission system;

II. *considering, moreover,*

that it is often desirable, in data transmission, to add an extra "parity" unit to allow for the detection of errors in received signals;

the possibility offered by this addition for the detection of faults in data terminal equipment;

the need to reserve the possibility of making this addition during the transmission itself, after the seven information units proper have been sent;

declares the view

that signals of the International Alphabet No. 5 code for data transmission should, in general, include an additional "parity" unit;

that the rank of this unit and, hence, the chronological order of the transmission in serial working should be the eighth of the combination thus completed;

III. *considering*

that, in start-stop systems working with electromechanical equipment, the margin of such equipment and the reliability of the connection are considerably increased by the use of a stop element corresponding to the duration of two unit intervals of the modulation;

that for transmissions over telephone circuits via modems installed on the user's premises, the latter must be able to use the connections at the highest possible practical rate in characters per second, and that in such a case a single-unit stop element leads to a gain of about 10% as regards this practical rate;

that, however, it does not appear that the production of electronic devices capable of working at will with start-stop signals having a stop element equal to one or two unit intervals should lead to costly complications and that such an arrangement can have the advantage of appreciably limiting the error rate without greatly reducing the practical efficiency of the connection;

declares the view

that in start-stop systems using combinations of the seven-unit alphabet normally followed by a parity unit, the first information unit of the transmitted combination should be preceded by a start element corresponding to condition A (space);

that the duration of this start element should be a one-unit interval for the modulation rate under consideration, at transmitter output;

that the combination of seven information units, normally completed by its parity unit, should be followed by a stop element corresponding to condition Z (mark);

that for start-stop systems using the seven-unit code on switched telephone networks, a two-unit stop element should be used with electromechanical data terminal equipments operating at modulation rates up to and including 200 bauds. In other cases, the use of a one-unit stop element is preferable. However, this is subject to a mutual agreement between Administrations concerned;

that similar situations when a one-unit stop element can be used may apply to leased circuits;

that the start-stop receivers should be capable of correctly receiving start-stop signals comprising a single-unit stop element, whose duration will be reduced by a time interval equal to the deviation corresponding to the degree of gross start-stop distortion permitted at receiver input. However, for electromechanical equipment

which must use a two-unit stop element (eleven-unit code signal) with a modulation rate of 200 bauds or less, receivers should be capable of correctly receiving signals with a stop element reduced to one unit;

IV. *considering, finally,*

that the direction of the parity unit can only be that of the even parity on the perforated tapes, particularly owing to the possibility of deletion (combination 7/15 of the alphabet) which causes a hole to appear in all tracks;

that, on the other hand, the odd parity is considered essential in the equipment which depends on transitions in the signals to maintain synchronism [in cases where combination 1/6 (SYNC) of the alphabet does not permit of an economical solution];

declares the view

that the parity unit of the signal should correspond to the even parity in links or connections operated on the principle of the start-stop system;

that this parity should be odd on links or connections using end-to-end synchronous operation;

that arrangements should be made when necessary to reverse the direction of the parity unit at the input and output of the synchronous equipment connected either to apparatus working on the start-stop principle or receiving characters on perforated tape;

that the detection of a character out-of-parity may be represented by:

a) reverse question mark (؟) graphic character or a representation of the capital letters SB (see ISO 2047 [2]) provided that these letters occupy a single character position on the screen or printer, and could have been entered by a single key stroke, recognizing it may be difficult to achieve a legible "SB" character from some matrix printers or displays where the characters are printed; or

b) a recording of the 1/10 (SUB) character on the tape or other storage medium, where provided

and that, where a SUB character occurs in a received transmission, or is presented to a DTE via a storage medium, e.g. paper tape, then the reaction should be as in a) and b) above.

References

[1] CCITT Recommendation *General structure of signals of International Alphabet No. 5 code for data transmission over public data networks*, Vol. VIII, Fascicle VIII.2, Rec. X.4.

[2] *Information processing – Graphical representations for the control characters of the 7-bit coded character set*, ISO Standard 2047, 1975.

Recommendation V.5

STANDARDIZATION OF DATA SIGNALLING RATES FOR SYNCHRONOUS DATA TRANSMISSION IN THE GENERAL SWITCHED TELEPHONE NETWORK

(former Recommendation V.22, Geneva, 1964;
amended at Mar del Plata, 1968, at Geneva, 1972 and 1976)

1 Data transmission by international communications carried on the general switched telephone network using a synchronous transmission procedure will be done with a specific mode of modulation, two- or multi-condition, and serial transmission (see Note 1). For synchronous data transmission on leased telephone-type circuits see Recommendation V.6.

2 The data signalling rates for synchronous transmission in the general switched telephone network will be:

600, 1200, 2400 or 4800 bits/s (see Note 2).

The users will choose among these rates, in accordance with their needs and the facilities afforded by the connection.